

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ
СІКОРСЬКОГО»**

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

"На правах рукопису"
УДК _____

«До захисту допущено»

Завідувач кафедри

_____ О.В. Коваль

(підпис)

(ініціали, прізвище)

“ _____ ” _____ 2019 р.

Магістерська дисертація

зі спеціальності - 122 Комп'ютерні науки
за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання
процесів та систем
на тему: Дослідження та розробка технологій агрегації інформаційних
джерел з урахуванням їх важливості

Виконав (-ла): студент (-ка) 6 курсу, групи ТМ-81мп

_____ Демидов Дмитро Володимирович

(прізвище, ім'я, по батькові)

_____ (підпис)

Науковий керівник к.т.н. Андрійчук О.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Рецензент к.т.н., с.н.с. ІПРІ НАНУ Каденко С.В.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

_____ (підпис)

Засвідчую, що у цій магістерській дисертації
немає запозичень з праць інших авторів без
відповідних посилань.

Студент _____
(підпис)

Київ - 2019

**Національний технічний університет України
“Київський політехнічний інститут ім. Ігоря Сікорського”**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти другий, магістерський

зі спеціальності - 122 Комп'ютерні науки

за спеціалізацією - Комп'ютерний моніторинг та геометричне моделювання процесів та систем

ЗАТВЕРДЖУЮ
Завідувач кафедри
Коваль О.В.
(прізвище, ініціали) (підпис)
«___» _____ 2019р.

**З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ ДИСЕРТАЦІЮ СТУДЕНТУ**

Демидов Дмитро Володимирович

(прізвище, ім'я, по батькові)

1. Тема дисертації Дослідження та розробка технологій агрегації інформаційних джерел з урахуванням їх важливості

Науковий керівник к.т.н. Андрійчук О.В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “04” 11 2019 року № 3812-с

2. Строк подання студентом дисертації 12.12.2019 р.

3. Об'єкт дослідження комп'ютерні інформаційні технології агрегації і аналізу інформації з інформаційних джерел

4. Предмет дослідження комп'ютерні інформаційні системи вилучення, парсингу та обробки з інформаційних джерел відповідно релевантності до пошукових запитів

5. Перелік питань, які потрібно розробити:

- проаналізувати існуючі засоби пошуку в інформаційних джерелах;
- дослідити роль користувачів та їх функції;
- дослідити можливості агрегації інформаційних джерел;
- спроєктувати роботу основних об'єктів модуля, представити алгоритми

їх роботи;

— розробка програмного продукту, що буде надавати можливість здійснити агрегації інформаційних джерел з урахуванням їх важливості.

6. Орієнтований перелік ілюстративного матеріалу презентація на тему «Засоби збору та обробки інформаційних джерел»

7. Орієнтований перелік публікацій тези доповіді XVI міжнародної науково-практичної конференції аспірантів, магістрів і студентів (Київ, 24-27 квітня 2018 р); тези доповіді II всеукраїнської студентської науково-практичної конференції «Наука XXI століття: виклики, пріоритети, перспективи досліджень» (22 березня 2018 р.).

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Основна частина	Андрійчук О.В., доц., к.т.н.		

7. Дата видачі завдання « 12 » вересня 2018 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання магістерської дисертації	строки виконання етапів магістерської дисертації	Примітка
1	Аналіз існуючих рішень	12.09.18 р.	
2	Збір інформації	01.10.18р. – 03.02.19р.	
3	Аналіз існуючих механізмів генерації програмного коду за заданим шаблоном	04.02 – 31.05.19 р.	
4	Аналіз метрик оцінки якості програмного коду	01.05 – 20.05.19 р.	
5	Аналіз вимог до аналізатора програмного коду	03.06 – 25.10.19 р.	
6	Моделювання схеми роботи майбутньої програми	02.09 – 25.10.19 р.	
7	Розробка архітектури програмного забезпечення	11.11 – 19.11.19 р.	
8	Розробка програмного застосунку	22.11.19 р.	
9	Оформлення документації	21.11- 29.11.19 р.	
10	Захист	16.12.19 р.	

Студент

(підпис)

(прізвище та ініціали)

Науковий керівник

(підпис)

(прізвище та ініціали)

РЕФЕРАТ

Структура й обсяг дипломної роботи.

Магістерська дисертація складається зі вступу, 6 розділів, висновку, переліку посилань з 37 найменувань і містить 9 рисунки, 22 таблиці. Повний обсяг магістерської дисертації складає 95 сторінок, з яких перелік посилань займає 4 сторінки.

Актуальність теми. Агрегація інформації - перетворення деталізованої інформації в пакети (агрегати) даних, для їх подальших аналізу. Особливо помітною стає необхідність корегування існуючої інформації для цільового дослідження об'єднуючи декілька інформаційних джерел. Складність полягає у тому, щоб вибрати найбільш важливу інформацію з різних джерел. Автоматизація цього процесу суттєво зменшить час розробки, а як наслідок, кошти витрачені на неї.

Мета дослідження полягає в удосконаленні процесу агрегації інформаційних джерел за рахунок урахування їх важливості.

Об'єктом дослідження є процес агрегації та аналізу інформаційних джерел.

Предметом дослідження є технології вилучення, парсингу та обробки інформаційних джерел відповідно до релевантності пошуковим запитам.

Для досягнення поставленої задачі були сформульовані наступні **завдання дослідження**, що визначили логіку дослідження та його структуру:

- проаналізувати існуючі засоби пошуку в інформаційних джерелах;
- дослідити роль користувачів та їх функції;
- дослідити можливості агрегації інформаційних джерел;
- спроектувати роботу основних об'єктів модуля, представити алгоритми їх роботи;
- розробка програмного продукту, що буде надавати можливість здійснити агрегацію інформаційних джерел з урахуванням їх важливості.

Наукова новизна одержаних результатів. Удосконалено спосіб проведення

агрегації інформаційних джерел з урахуванням їх важливості.

Практичне значення одержаних результатів роботи полягає в розробці програмного продукту, який підвищить ефективність збору інформації за допомогою інформаційних джерел.

Ключові слова. *АГРЕГАЦІЯ, ПОШУК ІНФОРМАЦІЇ, АГРЕГАЦІЯ ДАНИХ, Python.*

ABSTRACT

The structure and volume of the thesis.

Master's thesis consists of an introduction, 6 chapters, conclusion, list of references with 37 titles and contains 9 figures, 22 tables. The full range of master's thesis is 95 pages with a list of links takes 4 pages.

Topicality of the theme. Information Aggregation - the conversion of detailed information into data packets (aggregates) for further analysis. Particularly noticeable is the need to correct existing information for targeted research by combining multiple information sources. The difficulty is to select the most important information from different sources. Automating this process will significantly reduce development time, and as a result, the money spent on it.

The purpose and problems of research. The purpose of the study is to develop tools for searching, analyzing and aggregating for the subsequent processing of summary information.

The object of the study. The object of the study is computer information technology aggregation of information sources, given their importance.

The subject of the study. The subject of the study is computer information technology aggregation of information sources, given their importance.

To accomplish the task, the following research objectives were formulated, which determined the logic of the research and its structure:

- analyze existing search tools in information sources
- explore the role of users and their functions;
- explore the possibilities of aggregation of information sources;
- to design the work of the main objects of the module, to present algorithms of their work;
- development of a software product that will enable aggregation of information sources taking into account their importance. analyze the existing implementation of automation tools for refactoring software code;

Scientific novelty of the results.The way of aggregation of information sources with regard to their importance has been improved.

The practical significance of the results work is to develop a software product that will increase the efficiency of information gathering through information sources.

Keywords.*AGGREGATION, INFORMATION SEARCH, DATA AGGREGATION, Python.*

ЗМІСТ

Вступ.....	13
Перелін скорочень і умовних позначень	17
1. Роль інформаційних джерел у розвитку науково-технічному прогресу	18
1.1. Інформаційні джерела та їх види.....	19
1.1. Агрегація інформаційних джерел	22
Висновки до 1 розділу.....	25
2. Огляд існуючих програмних рішень для вибору та первинної обробки бібліографічних баз	27
2.1. Система управління бібліографічною інформацією ENDNOTE.....	27
2.2. Система управління бібліографічною інформацією ZOTERO.....	29
2.3. Система управління бібліографічною інформацією Mendeley	34
Висновки до розділу 2.....	37
3. Обґрунтування засобів розробки системи.....	39
3.1. Мова програмування Python	40
3.2. Середовище розробки.	41
3.3. Технологія Web Scraping.	43
3.4. Синтаксичний аналіз або парсинг	47
3.5. Мова розмітки сторінок HTML.....	50
3.6. Оболонка PyQt.	52
3.7. Система керування базами даних MySQL.....	56
3.8. Засоби обробки інформації.	59
3.9. Технології використані при програмній реалізації.....	61
Висновки до 3 розділу.....	63
4. Опис програмної реалізації	64
4.1. Структурна модель.....	64
4.2. Діаграма прицендентів.....	65
4.3. Фреймворк в Qt Designer	67
4.4. Обробка отриманої інформації	68
Висновки до 4 розділу.....	70
5. Робота користувача з програмою	71
5.1. Рекомендації до апаратного і програмного забезпечення.....	71
5.2. Робота користувача з програмою	72

5.3. Інтерфейс пошуку статей.....	73
Висновки до 5 розділу.....	75
6. Стартап проекту.....	76
6.1. Опис ідеї проекту	76
6.2. Технологічний аудит ідеї проекту	78
6.3. Аналіз ринкових можливостей запуску стартап-проекту	79
6.4. Розроблення ринкової стратегії проекту.....	86
6.5. Розроблення маркетингової програми стартап-проекту	89
Висновки до 6 розділу.....	92
Висновки	93
Список використаних джерел	96

ВСТУП

Інформаційні технології суттєво змінюють життя. Нові технології, пристрої, машини та сервіси, які роблять більше інформації, ніж будь-коли раніше, оточують усіх. Доступні інформаційні мережі та сервіси, які можуть з'єднати кожного та забезпечити нові можливості для продуктивності, навчання та розваг. Ці технології включають доступ до інформації через комп'ютери, Інтернет, телефони, пейджери, стільникові телефони, телевізори та громадські електронні кіоски.

Ці інформаційні технології включають в себе електронні методи спілкування та доступу до баз даних, які дозволяють нам купувати, купувати, оплачувати рахунки, отримувати книги для читання та складати плани та домовленості з дому. Вони роблять можливим телекомунікації та надають більше варіантів для ведення бізнесу вдома. Багато функцій роботи можна легко виконувати, коли надається доступ до необхідних інформаційних ресурсів та телекомунікаційних послуг. Доступ до мультимедійних навчальних матеріалів на компакт-диску чи всесвітній павутині (WWW) дає студентам досвід, який раніше ніколи не був можливим і доступний за допомогою комп'ютера в будь-якому місці. Курси та дипломні програми також пропонуються в електронному вигляді і дають можливість людям брати участь, не виходячи з дому чи громади.

Ці інформаційні технології також є чудовим джерелом розваг, таким як "серфінг в Інтернеті", щоб дізнатися та поділитися більше про хобі, місця та людей та грати в останні мультимедійні ігри. Поняття "потреба" поширюється на весь світ органічної та соціального життя, вказуючи, таким чином, на природний зв'язок між цими двома вищими формами руху матерії. Його розгорнуте значення пов'язано перш за все з конкретизацією уявлень про

суб'єкта діяльності, а отже, про носія потреб. У цій іпостасі можуть виступати будь-який біологічний організм, людський індивід, спільнота людей (сім'я, рід, плем'я, народ), соціальний шар або соціальна група в рамках цієї спільноти (клас, стан, нація, покоління, коло професіоналів), соціальний інститут, що функціонує в суспільстві (система освіти, держава та її органи і т. п.), суспільство як певна соціальна система, нарешті, людство в цілому.

Агрегація даних - це складання інформації з баз даних з метою підготовки комбінованих наборів даних для обробки даних.

На базі значень однієї або декількох групуючих змінних (змінних розбиття) можна об'єднати спостереження в групи (агрегувати) і створити новий файл даних, що містить по одному спостереженню для кожної групи розбиття.

Потенціал Інтернету для консолідації та маніпулювання інформацією має нове застосування в об'єднанні даних, також відоме як скраптування екрана. Інтернет надає користувачам можливість консолідувати свої імена користувачів та паролі чи PIN-коди. Така консолідація дозволяє споживачам отримувати доступ до широкого спектру захищених PIN-кодами веб-сайтів, що містять особисту інформацію, використовуючи один головний PIN-код на одному веб-сайті. Інтернет-провайдери облікових записів включають фінансові установи, біржових посередників, авіакомпанії, часті листівки та інші програми винагород, а також рахунки електронної пошти. Агрегатори даних можуть збирати облікові записи або іншу інформацію з визначених веб-сайтів, використовуючи PIN-коди власників облікових записів, а потім надавати інформацію про облікові записи користувачів на одному веб-сайті, керованому агрегатором, на вимогу власника облікового запису. Послуги агрегації можуть пропонуватися як окрема, так і спільно з іншими фінансовими послугами, такими як відстеження портфеля та оплата рахунків,

що надаються спеціалізованим веб-сайтом, або як додаткова послуга для посилення присутності в Інтернеті підприємства, створеного поза віртуальним світом. Багато відомих компаній, що мають Інтернет, мають визнання цінності надання послуги з агрегації для розширення інших веб-сервісів та залучення відвідувачів. Пропозиція послуги збору даних на веб-сайт може бути привабливою через те, що вона часто залучає користувачів сервісу на веб-сайт хостингу.

Створення навігатора такими ресурсами, доступними через глобальні інформаційні комунікації, дає змогу оптимізувати зусилля витрачені на науковий пошук, може бути корисним для проведення студентських та дисертаційних досліджень. Зібрання та представлення тематичних колекцій веб-ресурсів на сайтах бібліотек дозволяє користувачам в онлайн-режимі ознайомитися з інформацією, що значно розширює можливості доступу до неї, забезпечує оперативність та зручність використання, змістовне оновлення, збереження та редагування інформації.

Семантичний пошук позначає пошук із значенням, на відміну від лексичного пошуку, де пошукова система шукає буквальні збіги слів запиту або їх варіантів, не розуміючи загального значення запиту. Семантичний пошук має на меті підвищити точність пошуку шляхом розуміння намірів і контекстуального значення термінів, які вони відображаються в просторі даних для пошуку, в Інтернеті чи в закритій системі, для отримання більш релевантних результатів. Системи семантичного пошуку розглядають різні моменти, включаючи контекст пошуку, розташування, наміри, варіацію слів, синоніми, узагальнені та спеціалізовані запити, відповідність понять та запити на природній мові для забезпечення відповідних результатів пошуку.

Деякі автори розглядають семантичний пошук як сукупність методів отримання знань із добре структурованих джерел даних, таких як онтології та

XML, які можна знайти в семантичній мережі. Такі технології дозволяють формально сформулювати знання про домен на високому рівні виразності і можуть дати можливість користувачеві детальніше вказати їх наміри під час запитів. Пакет прикладних програм розробляється мовою програмування Python у програмному середовищі Pycharm Community з використанням бази даних Microsoft SQL Server 2017. Програмний комплекс має ряд базових можливостей для вирішення задачі пошуку та первинної обробки інформації з бібліографічних джерел.

ПЕРЕЛІК УМОВНИХ СКОРОЧЕНЬ

БД	База даних
ПЗ	Програмне забезпечення
HTML	Hyper Text Markup Language — Мова розмітки гіпертекстових документів
IDE	Integrated development environment — Інтегроване середовище розробки
UI	User Interface — Інтерфейс користувача
ПК	Персональний комп'ютер
XML	Extensible Markup Language — Розширювана мова розмітки
API	Application programming interface – Прикладний програмний інтерфейс
СУБД	Система управління базами даних

1. РОЛЬ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ У РОЗВИТКУ НАУКОВО-ТЕХНІЧНОМУ ПРОГРЕСУ

Обсяг інформації із звичайних баз даних та веб-джерел за останні кілька років експоненціально зростає. Як ефективно використовувати цей величезний обсяг інформації, стало проблемою для окремих людей та організацій. У той час як існуючий бізнес здійснює адаптаційні перетворення, щоб впоратися з динамікою цифрової економіки, з'являються нові технології та послуги, які допомагають інтегрувати та отримувати цінності з відповідних джерел інформації.

Це дослідження фокусується на певному типі сервісу, який агрегує інформацію з різних джерел за допомогою Інтернет-технологій. Ввівши поняття агрегації інформації, ми визначимо її основні характеристики та проілюструємо кожного із прикладів діючих постачальників послуг. Включення технологій для агрегування інформації буде обговорено коротко. Більшість існуючих досліджень моделей електронного бізнесу, як правило, охоплюють спектр видів бізнесу, не вистачаючи глибокого аналізу деталей, що визначають життєздатність та стійкість електронних підприємств. Ми скористаємося більш цілеспрямованим підходом і використовуватимемо тематичний випадок, що стосується бізнесу, в об'єднанні інформації, щоб продемонструвати, що ключ до успіху виходить від сукупної інформації та можливості надання електронних послуг з доданою вартістю.

Збірка інформації також викликає ряд політичних та регуляторних змін. З урахуванням тенденції інтеграції світової економіки агрегація інформації діятиме в усьому світі. Буде обговорено завдання формування ефективної

міжнародної політики, що сприяє зростанню електронної комерції.

1.1. Інформаційні джерела та їх види

Джерело інформації - це людина, річ або місце, з яких походить, виникає або отримується інформація. Джерела інформації можуть бути відомі як первинні або вторинні. Це джерело може потім повідомити людину про щось або надати знання про це. Джерела інформації поділяються на окремі окремі категорії, первинні, вторинні, третинні тощо.

Типи джерел інформації. Різні гносеології мають різні погляди щодо важливості різних видів джерел інформації. Емпіризм розглядає сенсорні дані як кінцеві джерела інформації, тоді як інші гносеології мають різні погляди (Kragh, 1989). Різні типи інформаційних джерел можна розділити на дві широкі категорії.

- Документальні джерела;
- Недокументальні джерела;

Первинні джерела - це оригінальні документи події чи відкриття, такі як результати досліджень, експериментів чи опитувань, інтерв'ю, листи, щоденники, юридичні документи та статті наукових журналів. Первинні джерела - це також записи подій, як вони вперше описані. Це можуть бути відеокасети, аудіозаписи або новини звітів очевидців.

Первинні джерела інформації - це перші опубліковані записи оригінальних досліджень та розробок або опис нової програми чи нової інтерпретації старої теми чи ідеї. Є оригінальні документи, що представляють нефільтровані оригінальні ідеї.

Вони складають останню доступну інформацію. Дослідник, який виробляє нову інформацію, може зробити її доступною для конкретної

громади за допомогою першоджерел. Часто це може бути єдиним наявним джерелом інформації. Первинні джерела - це неорганізовані джерела, якими досить важко користуватися, вторинні джерела допомагають нам їх використовувати. Це важливі джерела інформації. Тема сама по собі стає дисципліною, коли в цій галузі починають вироблятися незалежні першоджерела. Швидкість зростання дисципліни значною мірою залежить від кількості літератури, що випускається у формі первинних джерел, що повідомляють про розвиток у відповідній галузі.

Первинне джерело - це термін, що використовується в ряді дисциплін для опису вихідного матеріалу, який є найближчим до людини, інформації, періоду чи ідеї, що вивчається.

В історіографії первинним джерелом (його також називають першоджерелом) є артефакт, документ, запис чи інше джерело інформації, який був створений у досліджуваний час. Якщо створено людським джерелом, то джерело, яке має безпосереднє особисте знання про описувані події.

Він служить оригінальним джерелом інформації про тему. Подібні визначення використовуються в бібліотечній науці та інших сферах наукової діяльності.

У журналістиці першоджерелом може бути людина, яка безпосередньо знає ситуацію, або документ, створений такою людиною. Первинні джерела відрізняються від вторинних джерел, які цитують, коментують або ґрунтуються на первинних джерелах. Хоча відмінність не є різкою. "Первинне і вторинне - відносні терміни, джерела оцінюються первинними або вторинними відповідно до конкретних історичних контекстів і того, що вивчається".

Вторинні джерела пропонують аналіз або перегляд події або відкриття, описаних у первинних джерелах. Вони інтерпретують, пояснюють або

узагальнюють першоджерела. Деякі вторинні джерела використовуються для переконання читача. Вторинні джерела можна вважати менш об'єктивними. Приклади вторинних джерел включають: словники, енциклопедії, підручники, статті та редакції, які інтерпретують або рецензують науково-дослідні роботи.

Вторинні джерела інформації - це ті, які або складені або відносяться до первинних джерел інформації. Вихідна інформація була випадково модифікована, вибрана або реорганізована таким чином, щоб служити певній меті для групи користувачів. Такі джерела містять інформацію, упорядковану та організовану на основі певного певного плану. Вони містять організовані перепаковані знання, а не нові знання. Інформація, подана в первинних джерелах, стає доступною у більш зручній формі. За своєю суттю вторинні джерела легше та широко доступні, ніж первинні. Вони не лише надають переварену інформацію, але й служать бібліографічним ключем до первинних джерел інформації. Перші джерела з'являються першими, за ними йдуть вторинні. Важко знайти інформацію з первинних джерел безпосередньо. Тому в першу чергу слід звернутися до вторинних джерел, що призведе до конкретних первинних джерел.

Відповідно до уподобань джерел, було встановлено, що людські, ЗМІ та Інтернет - це три найпопулярніші жанри інформаційних джерел загалом. Тому ми спочатку уважно подивилися на розподіл наших вихідних даних у рамках основної схеми трьох жанрів. Тоді ми вказали три жанри на основі обсягу джерел у контексті соціальних запитань та запитань та розробили нову схему класифікації джерел.

Високий рівень розподілу джерел за жанрами був класифікований на чотири категорії: людина, книги, засоби масової інформації та Інтернет. Хоча книги не вважалися джерелом, що цитується в попередніх дослідженнях, ця категорія була знову додана до жанрового аналізу в цьому дослідженні завдяки

впізнаваній кількості джерел у нашому контексті соціальних запитань та питань. Жанрові категорії були визначені наступним чином.

Обсяг людських джерел такий самий, як визначений у попередньому розділі. Людські джерела вказують на знання, досвід, професію чи роботу постачальників джерел (відповідачів у контексті нашого дослідження відповідей Yahoo!) або відповідних третіх сторін. У джерелах книг вказуються переважно друковані рукописи чи публікації, але посилання або URL-адреси онлайн-книг також були включені до цієї категорії. Заголовок або автори книг були показані як джерела. Засоби масової інформації вказують джерела з інформацією про передачі, новини, журнали чи телебачення. Наприклад, ім'я широкомовної програми або веб-сайт тієї самої програми може бути вказано як джерело інформації. Таким чином, вона включає як Інтернет, так і офлайн інформацію про ЗМІ. Решта Інтернет-джерел, які не входять до жодного жанру людини, книг чи засобів масової інформації, були згруповані в Інтернет-джерела.

1.2. Агрегація інформаційних джерел

Агрегація інформації - це послуга, яка збирає відповідну інформацію з декількох джерел для забезпечення зручності та додаткової цінності, аналізуючи агреговану інформацію для конкретних цілей за допомогою Інтернет-технологій.

У більш широкому розумінні інформаційні посередники, такі як газети, журнали, професійні журнали та останнім часом, збільшує кількість веб-порталів - агрегатори інформації, оскільки всі вони збирають інформацію з багатьох джерел та поширюють її для зручного споживання. Але вони не включені до цього дослідження, оскільки вони, як правило, обслуговують

загальні потреби в інформації та не мають функціональних можливостей аналізу сукупної інформації на різних рівнях деталізації для конкретних цілей.

Подібно до визначення для агрегування інформації, визначення у цій статті зосереджено на наборі інформаційних посередників, щоб забезпечити глибокий аналіз питань, пов'язаних з бізнесом, технологіями та політикою. Це визначення стане зрозумілішим, коли ми обговоримо його унікальні характеристики та розгортаємо механізми його створення у наступних розділах.

В даний час величезна кількість інформації в Інтернеті зробила складним завданням пошуку конкретної інформації. Що стосується покупки в Інтернеті, пошукова система пропонує незначну допомогу у пошуку товару та конкурентному постачальнику, який перевозить конкретний товар. Наприклад, пошук "пальмового пілота" за допомогою найпопулярнішої пошукової системи Google повертає близько 483 000 URL-адрес! Буде копітким завданням відвідати кожен із цих сайтів та знайти відповідну інформацію.

Для вирішення цієї проблеми з'явилася порівняльна агрегація. Він отримує інформацію про набір атрибутів для товару / послуги, пропонує багатьма конкуруючими постачальниками, і нормалізує інформацію для змістовного порівняння. Багато стрілецьких та пошукових агентів, такі як mySimon.com та DealTime.com, пропонують послуги зі збору даних порівняння.

Атрибути, що порівнюються, залежать від товару / послуги. Наприклад, для книг агрегатор порівняння може отримати ім'я продавця, рейтинг продавця, назву та формат книги (наприклад, тверда обкладинка, аудіокасети тощо), доступність, ціна, загальна ціна після оподаткування плюс доставка та обробка тощо.

Для таких послуг, як бездротовий зв'язок, він може отримувати інформацію, таку як постачальник, назва плану, чи є акція, включені хвилини, щомісячна плата, тариф на великі відстані тощо. Щоб переконатися, що ця інформація є значущим, агрегатори повинні примирити смислові конфлікти між різними джерелами. Наприклад, DealTime.com може розпізнати невідповідності валюти та здійснювати правильні конверсії, використовуючи останній обмінний курс.

Для залучення клієнтів та підвищення лояльності клієнтів багато компаній пропонують зручні онлайн-сервіси, такі як банківська справа та брокерські послуги. Це, у свою чергу, значно збільшило вибір споживачів серед різних постачальників послуг. Власність тісних стосунків зі своїми клієнтами важлива для бізнесу, орієнтованого на сервісні послуги.

З точки зору споживачів, вони несуть тягар управління своїми відносинами з кількома постачальниками послуг. Людині властиво мати кілька банківських рахунків, брокерські рахунки та кредитні картки. Оскільки для кожного акаунта потрібен логін, це було б нудно і витрачає багато часу на управління цими обліковими записами. На щастя, поява агрегації відносин може допомогти клієнтам керувати кількома обліковими записами за допомогою одного входу. Агрегатор відносин може збирати інформацію від імені замовника та створювати різні корисні звіти. Зручність доступу до всієї інформації з одного місця допомагає встановити тісний зв'язок між агрегатором та його замовниками. У США було розроблено ряд агрегаторів фінансових рахунків, щоб допомогти клієнтам управляти різними рахунками, наприклад, Yodlee, VerticalOne та Corillian, щоб назвати лише декілька.

Усвідомлюючи потенційну загрозу втрати клієнтів та можливості перехресних продажів, багато фінансових установ пропонують своїм клієнтам

послуги з агрегування. Веб-портали, такі як AOL та Yahoo, також пропонують агрегацію акаунтів.

Загальноприйнято, що інформація в організації поширюється між інформаційними системами, якими керують різні відділи та відділення. Служба згуртування внутрішньоорганізаційних організацій може агрегувати відповідну інформацію з різних джерел для просування знань обмін та проведення аналізу фірмового рівня. Наведемо приклад такого типу агрегації у прикладі.

Внутрішньоорганізаційна агрегація також може бути використана як альтернатива до системної інтеграції та стандартизації в деяких великих організаціях. Наприклад, інтеграція між старими системами та новими може здійснюватися шляхом агрегації з мінімальними витратами на розробку. Аналогічна ситуація існує міжорганізаційною агрегацією, а також сукупністю сукупності внутрішньо- та міжорганізаційної інформації, що агрегується.

Висновки до 1 розділу

Як відомо обсяг інформації із звичайних баз даних та веб-джерел за останні кілька років експоненціально зростає. Для того, щоб надати змогу користувачу шукати та об'єднувати різні інформаційні джерела була спроектована ця система

Необхідні можливості, які забезпечує модуль, є:

- Введення ключових слів для пошукового запиту;
- Пошук робіт серед усіх даних в джерелі у порядку згідно їх пріоритетам;
- Видача списку результатів пошуку та завантаження даних у БД;

- Завантаження результатів та посилань на статті згідно з релевантністю;
- Проведення оптимізації результатів первинного пошуку за рахунок визначення найбільшої суми ваги більш важливих ключових слів;
- Видача оптимального результату відповідно до списку даних першого пошуку;
- Видача інфографіки оптимізації результату пошуку.

2. ОГЛЯД ІСНУЮЧИХ ПРОГРАМНИХ РІШЕНЬ ДЛЯ АГРЕГАЦІЇ ІНФОРМАЦІЙНИХ БАЗ

Інформація - один з найважливіших активів сучасної організації. Обсяги даних, які зберігають і обробляють сучасні організації (навіть не найбільші), вимірюються сотнями гігабайт і продовжують зростати. Зростає не тільки кількість інформації, зростає її ціна. Вартість інформації, що зберігається в комп'ютері в десятки, а то і в сотні разів перевищує вартість обладнання. Навряд чи когось треба сьогодні переконувати в необхідності систем резервного копіювання та відновлення файлів - на існують істотні, часом багатомільйонні, збитки від втрати даних. Тепер системи зберігання даних стали широко поширеним товаром, який повинен відповідати потребам клієнта, володіти певними споживчими характеристиками і служити інструментом вирішення унікальних завдань щодо збереження інформації.

2.1. Система управління бібліографічною інформацією ENDNOTE

EndNote - комерційна система управління бібліографічною інформацією, що застосовується для управління посиланнями і бібліографією, що дозволяє відформатувати їх згідно з численними стандартами цитування. Компанія-виробник - Clarivate (в даний час, раніше - Thomson Reuters).

Робота з цитатами з тексту, особливо якщо вона виконується вручну, забирає багато часу і сил. Припустимо, необхідно додати цитату на початку тексту, це означає, що потрібно оновлювати всю нумерацію цитат.

Нарешті, після довгої монотонної і виснажливої роботи з текстом, стаття

готова для подачі в журнал, нумерація і формат вивірені, але стаття отримує відмову в публікації, це значить потрібно починати все практично, спочатку, і переробляти формат кожної цитати. З огляду на, що різні журнали мають різні вимоги до оформлення цитат як в тексті, так і в списку літератури, виправлення манускрипту в процесі роботи вимагають все нових цитат, а це значить потрібно виправляти весь список заново, ця робота забирає дні, а іноді і тижні, автори допускають помилки в нумерації цитування. На щастя, існує спосіб цей процес автоматизувати і ніколи більше не замислюватися про нумерацію і форматі ваших цитат, передоручивши роботу Endnote.

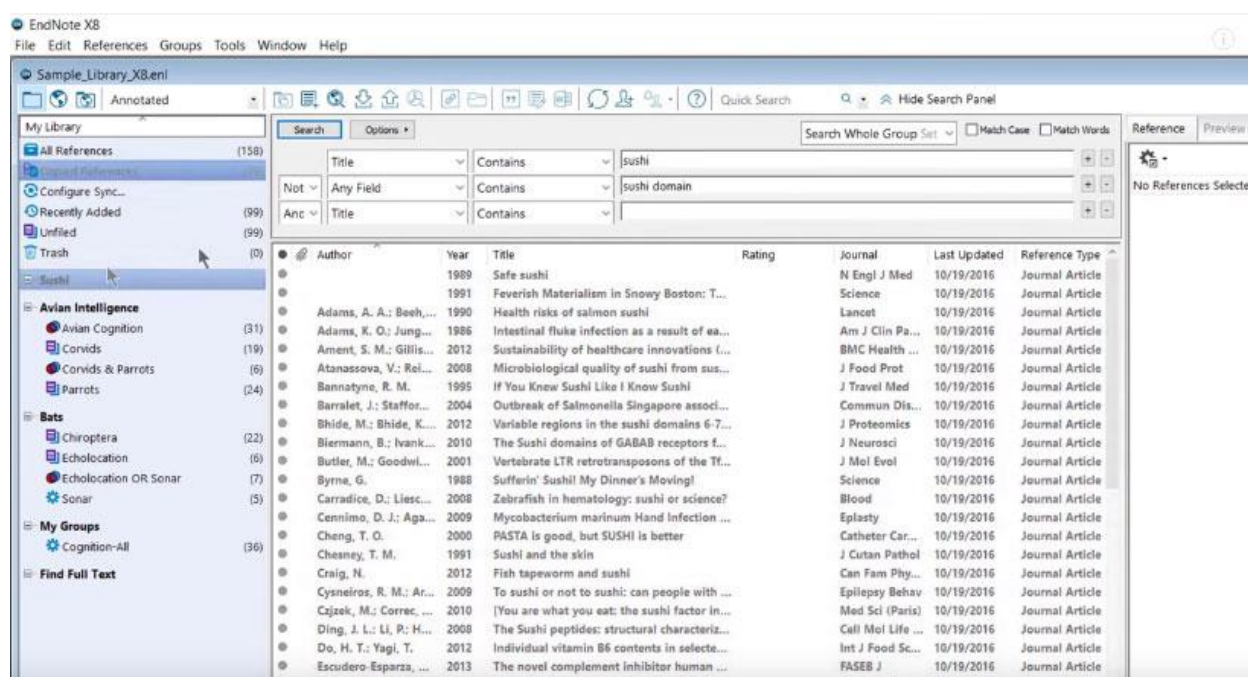


Рисунок 2.1 — Інтерфейс програми EndNote

Програма інформацію про статті отримує з он-лайн каталогів, за замовчуванням налаштовано з'єднання з Library of Congress, LISA, Pubmed і Web of Science. Можна додати і інші, але, як правило, встановлених досить. Зрозуміло, для завантаження нових цитат програмі буде потрібно підключення до інтернет. Тепер, підключившись до Pubmed, давайте знайдемо статтю, яку

ви б хотіли процитувати. Система пошуку дозволяє шукати по масі параметрів, але сам пошук реалізований достатньо не гнучко, на точних збіги, наприклад, виникають складнощі з грецькими символами дефісами і так далі. Чи не буде точного збігу, пошуковий запит прийде порожнім. Тому якщо пошук в Endnote не дає результатів це не означає, що статті в каталогах немає, потрібно змінити умови. Наприклад, шукати по імені автора, року публікації і назвою журналу, ніж назві статті, або використовувати перші кілька слів назви, замість повної назви.

Найбільш ефективна для мене схема роботи така: Написати фрагмент тексту, до якого потрібна цитата, потім в Google знайти підтверджує це твердження статтю або книгу, (часто відбувається і навпаки) потім скопіювати назву або автора в Endnote і додати цитату в текст.

Якщо стилю журналу, в який ви хочете подати статтю в списку стилів не виявилось, то забиваєте в пошук% ім'я журналу% Endnote style і качаєте ens файл.

2.2. Система управління бібліографічною інформацією ZOTERO

Zotero - це безкоштовна бібліографічна програма, яка управляє посиланнями (цитатами), які ви збираєте з баз даних, каталогів бібліотек, веб-сайтів або створюєте вручну. Програма автоматично формує збережені цитати в будь-якому стандартному форматі. Zotero здатний зберігати повнотекстові статті, створюючи знімок та / або веб-посилання будь-якої веб-сторінки. Zotero також пропонує можливість зберігати та синхронізувати зібрані посилання на сервері на базі Інтернету.

Zotero призначений для роботи зі складними документами. Він дозволяє планувати роботу з використовуваними джерелами, проводити їх опис,

реферування і готувати публікації. Zotero це плагін до браузера Firefox. Для роботи з Zotero браузер повинен бути відкритим, з тим, щоб отримати доступ до Ваших добірок. Zotero не вимагає для роботи постійного підключення до інтернету - частина функцій Zotero, таких як: Нотатки, Пошук, Дії буде працювати в автономному режимі. Однак, Ви не зможете переглядати деякі пункти Ваших добірок: бібліографічні посилання, цілі документи, файли, записи та інші об'єкти, наприклад, зображення веб-сторінок. Для відкриття вікна Zotero необхідно натиснути іконку в нижньому правому куті вікна браузера.

Основний логічною одиницею в Zotero є документ. Документи описуються по типам (книга, лист, фільм, інтерв'ю, карта і т.п.), атрибутам (рік видання, сторінка, URL, випуск, видавець, ISBN, мова і т.п.), тегам (ключовими словами). документи можуть містити замітки (це зручно для формування виписок і цитат), додатки (Локальні файли) і бути пов'язаними з іншими документами. для об'єднання документів по одному напрямку літературного пошуку використовуються добірки.

Zotero є плагін (розширення) для браузера Mozilla Firefox для збору і управління бібліографічною інформацією, а також напівавтоматизованого оформлення бібліографічних посилань і списків. Розробником програмного продукту є компанія Center for History and New Media. Програмний продукт випускається, як розширення браузера Mozilla Firefox, під ліцензією GNU, що робить його безкоштовним у використанні. Дистрибутив можна скачати з сайту www.zotero.org, на початку травня 2011 р останній випуск системи управління був Zotero 2.1.6 для Mozilla Firefox 3.6 і 4.0. Якщо на комп'ютері не встановлено браузер Mozilla Firefox, його дистрибутив може бути завантажений з офіційного сайту Mozilla. Необхідно врахувати, під керуванням якої операційної системи працює комп'ютер і вибрати відповідну

версію програмного продукту. Інсталяція на комп'ютерах під управлінням операційної системи Microsoft Windows здійснюється запуском на виконання файлу дистрибутива Firefox Setup.exe. Після установки браузера необхідно зайти на сторінку www.zotero.org і натиснути на кнопку Download. При цьому повинна розпочатися не скачування файлу дистрибутива, а установка розширення, про що користувач буде повідомлено відповідному діалоговому вікні. Як тільки Firefox встановить Zotero й увімкнеться, можна зайти в розширення, натиснувши на іконку Zotero в правому нижньому кутку вікна Firefox. При цьому відкривається його власне вікно.

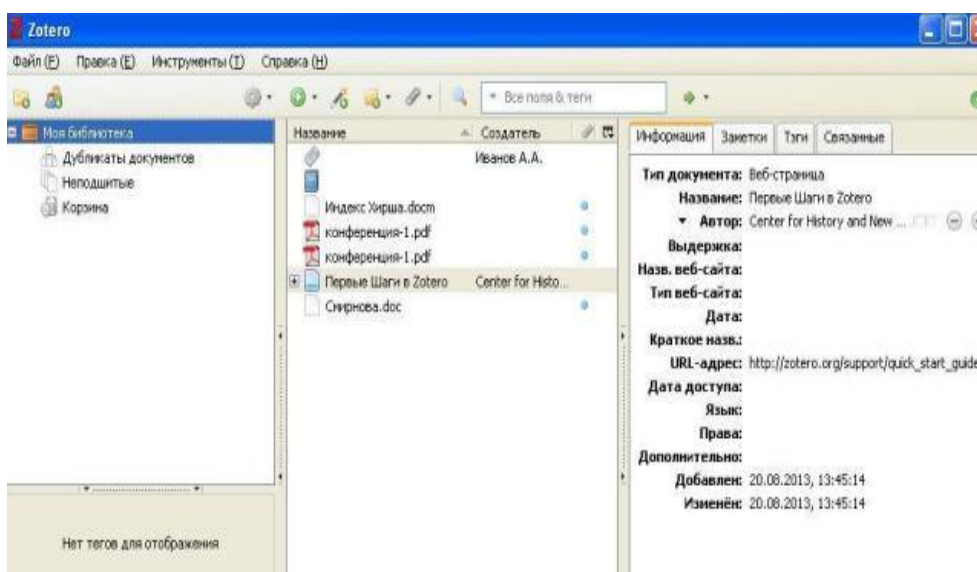


Рисунок 2.2 — Інтерфейс програми Zotero

Zotero - це менеджер цитування. Він спроектований для зберігання, управління і цитування бібліографічних посилань. У Zotero кожна з таких посилань називається елементом.

У вікні на лівій комбінованій панелі розташована папка Моя Бібліотека, яка містить всі елементи. Натисканням відповідної кнопки можна створити нову колекцію- папку, в якій можна розміщувати елементи, що стосуються

окремого проекту або предметної області. Колекції можуть містити субколекції. Елементи до колекціях можуть дублюватися, але це не супроводжується дублюванням файлів занесених в інформаційну систему файлів документів, а лише їх назв (імен).

Основний спосіб наповнення власної інформаційної системи - створення і наповнення колекцій. Zotero дозволяє поміщати один елемент в кілька колекцій одночасно. При цьому визначані будуть операції з елементами в основній папці Моя Бібліотека, які поширюються на всі однойменні елементи у всіх існуючих колекціях. І навпаки, операції всередині колекції обмежені межами її самої.

Рядок кнопок містять управління, що дозволяють виконати основні операції по акумуляції, структурування і вилучення бібліографічної інформації. Імпорт документів здійснюється за допомогою вибору операції в випав-дающем меню, що викликається натисканням на кнопку Дії. Підтримується імпорт даних з файлів практичних форматів електронної бібліографії: MODS, BibTeX, MAB2, MARC, RDF, RIS, Refer / BiblX.

Інсталяція браузера Mozilla Firefox. Zotero є плагін (розширення, надбудову) браузера Mozilla Firefox. Даний браузер належить до числа вільно поширюваного відкритого програмного забезпечення. Дистрибутив програми може бути безкоштовно просмотрів за браузера. Скачаний файл Firefox Setup.exe необхідно запустити на виконання і, при необхідності, вказати логічний диск і каталог для установки програми. Користувач може, але не зобов'язаний реєструватися як користувач програмного забезпечення.

Інсталяція плагіна Zotero для MS Office Word. Даний плагін в дійсності є доповнення не текстового процесора MS Office Word, а браузера Mozilla Firefox. Його установка здійснюється аналогічним чином з відповідної сторінки. Плагін надає можливість інтегрувати можливості Zotero з

управління бібліографічними даними з можливостями текстового процесора за їх уявленням (перш за все, оформлення посилань і бібліографічних списків).

Реєстрація користувача на сайті Zotero не є обов'язковою, але надає можливість використання такого в якості віддаленого сервера для зберігання своєї бібліотеки і організації колективного або віддаленого використання бібліотек і окремих збірок (колекцій). Безкоштовно надається до 100 Мб дискового простору.

Сортування елементів (інформаційних записів) можлива за всіма основними параметрами записів, в тому числі за назвою, автору, типу інформаційного джерела, датою, році, видавництву, мови, наявності файлу першоджерела і т.д.

Пошук засобами Zotero здійснюється тільки всередині бібліотеки по всіх наявних колекцій. При великому числі він може бути уповільненим через потребу у великій обчислювальній потужності комп'ютера.

Всі файли, наявні в бібліотеці Zotero, зберігаються в папці профілів браузера Mozilla Firefox (якщо не вказано інше місце розташування для зберігання на вкладці завдання «розширених» параметрів). Кожен файл знаходиться в окремому підкаталозі в папці зберігання, названої довільним поєднанням восьми символів. З цієї причини недоцільно намагатися шукати ці файли файлових менеджерів, відкривати ці файли слід безпосередньо через Zotero. Вибір в Zotero елемента і опції Перегляд в правій колонці веде до відкриття документа засобами браузера.

Створення посилання найпростіше здійснити механічним перетягуванням потрібного запису з вікна Zotero в потрібне місце текстового документа. Вставка цитати здійснюється аналогічним чином, але при натиснутій клавіатурній клавіші Shift.

Створення списку використаних джерел задається з меню, що випадає,

що відкривається при натисканні правої клавіші маніпулятора під час знаходження курсору в тексті. Додатковими параметрами є завдання і модифікація формату генеруються бібліографічних списків. Zotero надає можливість використання великої кількості встановлених шаблонів оформлення бібліографічних списків, але, найчастіше, доводиться виконувати роботу по остаточному оформленню таких списків.

Синхронізація дозволяє резервувати (зберігати) в мережевому доступі на сервері елементи бібліографії, але не файли. Віддалене зберігання файлів здійснюється в так званому Сховище Zotero.

2.3. Система для управління бібліографічною інформацією Mendeley

Mendeley - це унікальне рішення, що поєднує в собі reference manager (засіб збору і каталогізації наукових статей для подальшої підготовки пристатейних списків) і наукову соціальну мережу, яка дозволяє організовувати персональну наукову бібліотеку, спільно працювати над статтями, а також знаходити однодумців і вивчати тренди сучасних досліджень .

Mendeley - це настільна та веб-програма, розроблена компанією Elsevier для управління та обміну науково-дослідницькими матеріалами, виявлення дослідницьких даних та співпраці в Інтернеті. Він поєднує в собі Mendeley Desktop, PDF та довідкову програму керування, доступну для Windows, macOS (Sierra і High Sierra більше не підтримуються) та Linux. Mendeley також доступний для Android і iOS, а також Mendeley Web - онлайн-соціальну мережу для дослідників.

Mendeley вимагає, щоб користувач зберігав усі базові дані цитування на

своїх серверах, зберігаючи копії документів на власний розсуд. Після реєстрації, Менделі надає користувачеві 2 Гб вільного простору в Інтернеті, що можна оновлювати за ціною.

Менделі доступний або як преміальна платна версія або як базовий варіант, який є безкоштовним, але вимагає реєстрації.

Mendeley Desktop, розроблений на Qt, працює на Windows, Mac (MacOS Sierra і High Sierra офіційно не підтримується) і Linux.

Mendeley надає наступні можливості:

- резервне копіювання та синхронізація на кількох комп'ютерах та в приватній мережі;
- переглядач PDF з замітками, виділення тексту та повноекранне читання;
- повнотекстовий пошук по статтях;
- розумна фільтрація, тегування та автоматичне перейменування файлів PDF;
- цитати та бібліографічні посилання в Microsoft Word, OpenOffice.org та LibreOffice;
- імпорт документів та наукових статей із зовнішніх веб-сайтів (наприклад, PubMed, Google Scholar, Arxiv) через букмарклет браузера;
- Експортувати BibTeX / синхронізувати файл;
- Функції соціальних мереж (новинні канали, коментарі, сторінки профілю тощо);

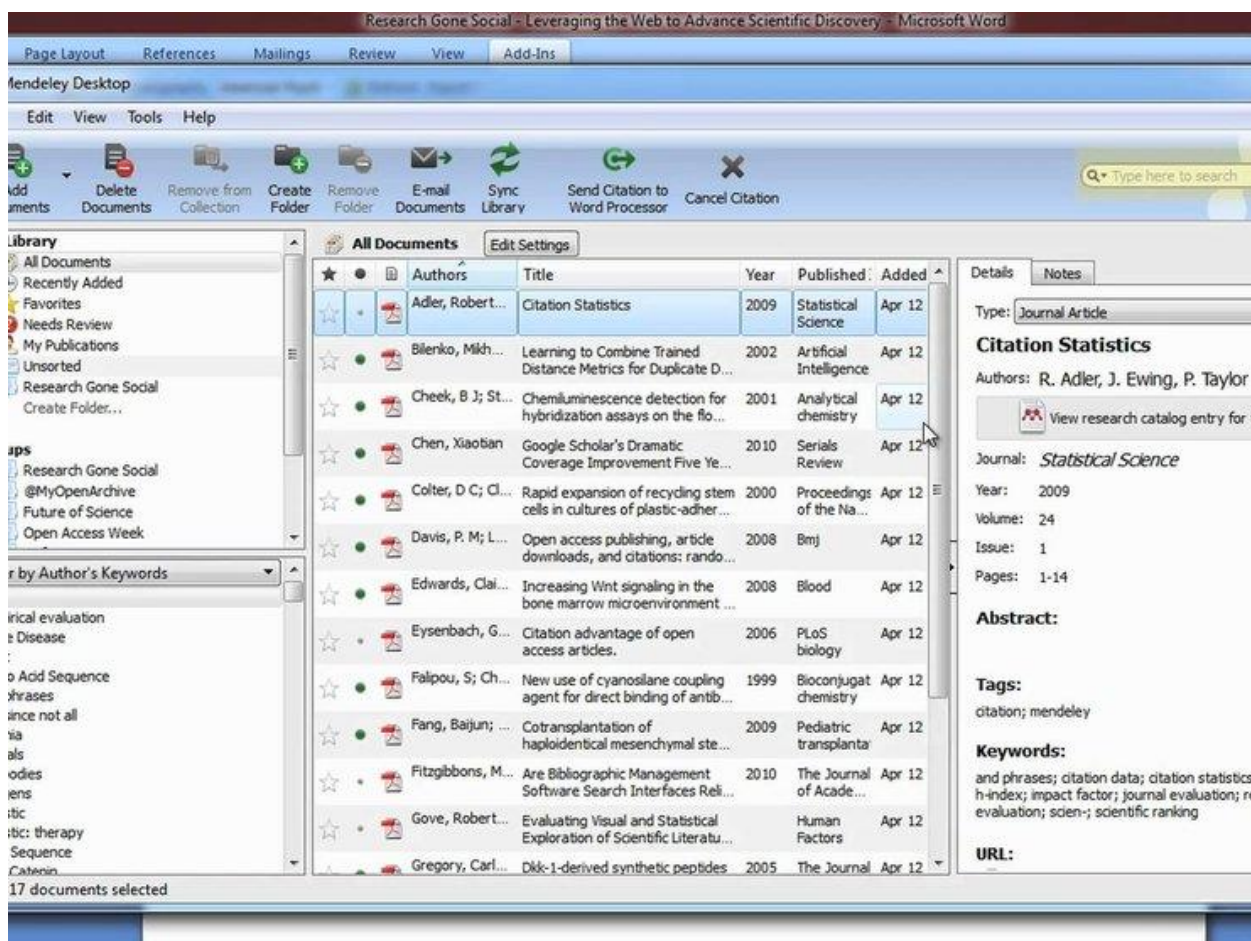


Рисунок 2.3 — Інтерфейс програми Mendeley

Невід'ємною частиною наукової діяльності є робота з текстами наукових статей, доповідей, монографій та інших наукових публікацій, виданих як традиційним способом на папері, так і в електронному вигляді (так званий електронний ресурс). Робота з документами в електронному вигляді дає ряд додаткових можливостей, як, наприклад, автоматизація пошуку інформації та її систематизація. У процесі наукової роботи та підготовки обстоювання публікацій дослідник стикається з такими завданнями, як створення колекцій публікацій, систематизація та складання їх бібліографічного опису; створення коротких рефератів, анотацій та позначок при аналізі результатів, викладених в наукових публікаціях. При написанні власних наукових робіт перед вченим виникає задача цитування раніше відомих результатів і складання переліків

посилань. Одним з інструментів, що дозволяє автоматизувати рішення зазначених завдань, є система Mendeley. Система Mendeley є комбінацією локальної програми Mendeley Desktop, яка є системою управління базами даних (СКБД), і веб-сервісу.

Доступ до системи можливий як на безкоштовній, так і на платній основі. Різні способи доступу відрізняються обсягом доступного місця для зберігання файлів на сайті, кількістю груп, які можна організувати, і кількістю членів у кожній групі. У разі безкоштовного доступу в системі виділяється 1 Гбайт для зберігання документів, по 500 Мбайт для закритих і публічних колекцій. При необхідності зберігання в режимі он-лайн великого обсягу документів можливе використання зовнішніх веб-сервісів, наприклад, SugarSync, який дозволяє зберігати до 5,5 Гбайт даних з автоматичною синхронізацією з локальними комп'ютерами.

Mendeley дозволяє об'єднати інформацію на персональному комп'ютері користувача і в Інтернеті, надає кошти пошуку та інші можливості для комунікації, обміну та обговорення. При цьому системою можна користуватися без прив'язки до конкретного комп'ютера. Доступ в систему можливий за допомогою веб-браузера, за допомогою програм-клієнтів, які встановлюються на комп'ютері, що працює під управлінням операційних систем Mac, Windows і Linux. Можливі також доступ з мобільних пристроїв, таких як iPhone і iPad. При установці клієнтського програмного забезпечення на декількох комп'ютерах можлива автоматична синхронізація даних на цих комп'ютерах за допомогою мережі Інтернет.

Висновки до розділу 2

Зараз наявна досить велика кількість програм управління

бібліографічною інформацією, що надають можливість проводити пошук та зручне користування можливостями перегляду та збереження наукових статей та. Всі вони мають дуже схожий функціонал, проте у кожного є свої особливі набори компонентів та інструментів, мають різний інтерфейс, деякі розроблені для встановлення на власний сервер, деякі представляють собою онлайн-сервіси, також різняться і цінову політику та правила, за якими вони розповсюджуються. Розглянуті сервіси є найбільш популярними з безлічі подібних, проте загалом всі функції для аналізу тексту, які необхідно використовувати являються платними.

3. ОБҐРУНТУВАННЯ ВИБОРУ ЗАСОБІВ РЕАЛІЗАЦІЇ ПРОГРАМНОГО КОМПЛЕКСУ

При розробці програмного забезпечення програміст має вибрати такі технології й засоби, які б полегшили його роботу, надавши всі необхідні інструменти для реалізації завдання. Правильний вибір технологій і засобів програмної реалізації впливає на час розробки, якість, надійність і швидкість роботи продукту.

Для створення програмного комплексу було обрано такі засоби:

- середовище розробки програмного забезпечення — PyCharm 3.1.2;
- мова програмування — Python;
- для розмітки сторінок — HTML і каскадні таблиці стилів CSS;
- для створення бази даних і роботи з нею — СКБД MySQL.

Програмний комплекс працює на операційних системах на усіх операційних системах, на яких встановлена мова програмування Python 3.0.

Тестування програмного комплексу пошуку та первинної обробки інформації виконано на інтегрованому середовищі розробки Pycharm 2018.1.2. Основним середовищем розробки була середовище розробки PyCharm, яка є однією з найпопулярніших для Python середовища розробки.

Програмний інструмент був розроблений з використанням мовних функцій Python і управління даними SQL.

3.1. Мова програмування Python

Python - інтерпретована, об'єктно-орієнтована мова програмування високого рівня з динамічною семантикою. Високорівневі вбудовані структури даних у поєднанні з динамічним набором тексту та динамічним зв'язуванням роблять її дуже привабливою для швидкого розвитку додатків, а також для використання як мови сценаріїв або клею для з'єднання існуючих компонентів разом. Простий, легкий у вивченні синтаксис Python підкреслює читабельність і, таким чином, зменшує витрати на обслуговування програми. Python підтримує модулі та пакети, що заохочує модульність програми та повторне використання коду. Інтерпретатор Python та велика стандартна бібліотека доступні у вихідній чи двійковій формі безкоштовно за всіма основними платформами, і їх можна вільно поширювати.

Портований Python і працює практично на всіх відомих платформах - від мейнфреймів до КПК. Є порти під Microsoft Windows, всі версії UNIX (включаючи FreeBSD та GNU / Linux), Plan 9, Mac OS і Mac OS X, iPhone OS 2.0 і пізніших версій, Palm OS, OS / 2, Amiga, AS / 400 і навіть OS / 390 [s], Symbian і Android.

Зі старінням операційної платформи мова перестає підтримувати їх в основній галузі. Наприклад, версія 2.6 більше не підтримується в Windows 95, Windows 98 і Windows ME. Однак ці платформи можуть підтримувати більш ранні версії спільноти Python, активно підтримують версії 2.3 на Python.

Однак, на відміну від багатьох систем, розміщених на всіх основних платформах, Python підтримує ці конкретні технології платформи (наприклад, Microsoft COM / DCOM). Крім того, існує спеціальна версія віртуальної машини Python для Java - Jython, так що перекладачі працюють на будь-якій системі, яка підтримує Java, як класи Java можна використовувати безпосередньо з Python, також написані на них. Крім того, деякі проекти

пропонують інтеграцію з платформою Microsoft.NET, найбільш важливі з яких - Ironpython і Python.Net.

3.2. Середовище розробки

PyCharm - інтегроване середовище розробки для мови програмування Python. Надає інструменти аналізу коду, графічний відладчик, інструмент для запуску тестування компонентів і підтримує веб-розробку на Django. PyCharm розроблений чеською компанією JetBrains на базі IntelliJ IDEA.

PyCharm працює в операційних системах Windows, Mac OS і Linux.

варіанти:

- Аналіз статичного коду, підсвічування синтаксису і помилки.
- Як перейти від однієї проектами і необробленим кодом: відображення файлової структури проекту, швидке переміщення між файлами, класами, методами і методами.
- Рефакторинг: перейменування, вилучення методу, введення змінної, введення константи, збільшення і зменшення методу і т. Д.
- Інструменти для веб-розробки, що використовують середу Django
- Вбудований відладчик для Python
- Інтегровані інструменти тестування Unity
- Розроблено за допомогою Google App Engine

Підтримка системи управління версіями: загальний користувальницький інтерфейс для Mercurial, Git, Subversion, Perforce і CVS з підтримкою списків змін і злиттів.

PyCharm Professional Edition має кілька варіантів ліцензування, які відрізняються функціональністю, вартістю і умовами використання. PyCharm Professional Edition безкоштовно для освітніх установ і проектів з відкритим

вихідним кодом. Існує також безплатна версія Community Edition Edition с обмеженим тиражом, розповсюджувана по ліцензії Apache 2.

можливості:

Python Profiler. Візьміть повний контроль над Вашим кодом з абсолютно новою інтеграцією Python Profiler. Підтримує Yappi і cProfile.

Вбудований Debugger. PyCharm відладчик тепер включає в себе відладчик для легкого огляду змінних, параметрів функцій і об'єктів прямо в редакторі під час налагодження програми.

Interactive matplotlib. Matplotlib інтерактивний режим тепер працює в обох Python і Debugger консольях.

Ігнорування файлів бібліотеки. Отладчик поліпшується з новою опцією «Ігнорувати файли бібліотеки». Використовуйте його, щоб відладчик працював тільки всередині Вашого коду в разі, якщо код викликає виключення в модулі бібліотеки.

Step into My Code. Зосередьтеся на Вашому коді - відладчик пройде крок за кроком тільки код проекту, а не всієї бібліотеки.

Покращена навігація. Тепер є можливість перейти від відладчика відразу до змінних в коді, де визначаються об'єкт або його тип.

manage.py Tool. PyCharm 4.5 вводить істотно перероблений manage.py інструмент для проектів Django, з новим призначенням для користувача інтерфейсом, автозавершенням для manage.py завдань, швидким пошуком документації та історії команд. Django 1.8 Code Insight також був поліпшений.

Bulk Move Refactoring. Застосування нового рефакторінга дозволить надійно переміщати кілька об'єктів в модулі, наприклад, функцій, класів і змінних в інші файли.

Конвертація Module / Package. Завдяки цій новій функції рефакторінга, Ви можете сміливо перетворити існуючий модуль Python в пакет, або навпаки.

IPython Notebook. IPython Notebook інтеграція була значно поліпшена з новою IPython Notebook консоллю і численними виправленнями помилок.

Scratch Files. Тепер Ви можете працювати з тимчасовими файлами Python Scratch Files, які не є частиною проекту.

3.3. Технологія Web Scraping

Web-scraping, збіркою веб-сторінок або видобуванням веб-даних є видалення даних, що використовується для вилучення даних з веб-сайтів. Програмне забезпечення для веб-скребків може отримати доступ до World Wide Web безпосередньо за допомогою протоколу передачі гіпертексту або через веб-браузер. Хоча веб-скребком можна зробити вручну користувачем програмного забезпечення, цей термін зазвичай стосується автоматизованих процесів, що виконуються за допомогою бота або веб-сканера. Це форма копіювання, в якій конкретні дані збираються та копіюються з Інтернету, як правило, в центральну локальну базу даних або електронну таблицю для подальшого пошуку чи аналізу.

Копіювання веб-сторінки включає завантаження та вилучення з неї. Отримання - це завантаження сторінки (яку браузер робить під час перегляду сторінки). Тому веб-сканування є основним компонентом веб-скребком, щоб отримати сторінки для подальшої обробки. Після того, як вирушає, видобуток може відбутися. Зміст сторінки може бути проаналізовано, обшуковано, переформатовано, його дані скопійовані в електронну таблицю тощо. Веб-скребки, як правило, беруть щось із сторінки, щоб використовувати його для інших цілей деінде. Прикладом може бути пошук і копіювання імен та телефонних номерів або компаній та їх URL-адрес до списку (контактне сканування).

Web-scraping використовується для контактного скрепінга, а також як складова додатків, що використовуються для веб-індексації, видобутку та видобутку даних, онлайн моніторингу зміни цін та порівняння цін, перегляду товару (для перегляду конкуренції), збирання списків нерухомості, даних про погоду моніторинг, виявлення змін у веб-сайті, дослідження, відстеження присутності та репутації в Інтернеті, веб-машининг та інтеграція веб-даних.

Веб-сторінки побудовані за допомогою текстових мов націлення (HTML та XHTML), і часто містять велику кількість корисних даних у текстовій формі. Однак більшість веб-сторінок призначені для кінцевих користувачів, а не для зручності автоматичного використання. Через це були створені набори інструментів, які б'ють веб-вміст. Web-scraping - це інтерфейс додаткового програмування (API) для вилучення даних з веб-сайту. Такі компанії, як Amazon AWS та Google, надають кінцевим користувачам безкоштовні кошти, послуги та публічні дані для веб-скребків.

Новіші форми Web-scraping включають прослуховування каналів даних з веб-серверів. Наприклад, JSON зазвичай використовується як механізм зберігання транспортних засобів між клієнтом та веб-сервером.

Існують методи, які деякі веб-сайти використовують для запобігання веб-скребком, наприклад, виявлення та заборони ботів від сканування (перегляду) своїх сторінок. У відповідь на це існують веб-скребкові системи, які спираються на використання методів аналізу DOM, комп'ютерного бачення та обробки природних мов, щоб імітувати пошук людей, щоб дозволити збирати вміст веб-сторінок для автономного синтаксичного аналізу.

Web-scraping - процес автоматичного збору інформації з World Wide Web. Ця область активного розвитку із загальною метою Semantic Web Vision - це амбітна ініціатива, яка як і раніше вимагає проривів в обробці текстів, семантичному розумінні, штучний інтелект і взаємодії людини і комп'ютера.

Сучасні веб-рішення, починаючи від skrapinhu ad hoc, вимагають зусиль людини для повністю автоматизованих систем, які трансформують цілі веб-сайти в структуровану інформацію з обмеженнями.

Ручне копіювання і заміна іноді навіть найкращої технології Web-scraping paste - це не ручна оцінка людей і копіювати-вставку, і іноді це може бути єдиним прийнятним рішенням, коли веб-сторінки задають бар'єри, встановлені (CAPTCHA), щоб запобігти автоматичну обробку.

Шаблони (регулярні вирази) - простий, але ефективний підхід до вилучення інформації з веб-сторінок. Використовує пошук текстових шаблонів (регулярних виразів), які підтримують багато мов програмування.

HTTP-програмування. Статичні і динамічні веб-сторінки можна видалити, помістивши HTTP-запити на віддалений веб-сервер через програмування сокетів.

Аналізатори HTML: на багатьох веб-сайтах динамічно генеруються великі колекції сторінок з просто структурованих джерел, таких як бази даних. Ця ж категорія зазвичай кодується на цих сторінках, використовуючи загальний скрипт або шаблон. В Data Mining програма, яка розпізнає шаблони, такі як певні джерела інформації, витягує її вміст і перетворює його в реляційну форму, звану оболонкою. Алгоритм генерації обгортки передбачає, що вхідні сторони в індукційної оболонці відповідає загальній схемі і що її можна легко ідентифікувати по відношенню до загального системного URL. Крім того, деякі напівструктуровані мови запитів даних, такі як XQuery і HTQL, можуть використовуватися для аналізу та вилучення HTML-сторінок і перетворення вмісту на сторінці.

Аналіз DOM: впровадження в повний веб-браузер, наприклад Internet Explorer або Mozilla, програми можуть отримувати динамічний контент, створюваний клієнтськими сценаріями. Ці елементи управління дозволяють

браузерам аналізувати веб-сторінки в дереві DOM, до яких додатки можуть відокремлювати сторінки.

Програмне забезпечення Web-scraping Існує безліч програмних інструментів, які можна використовувати для налаштування рішень веб-скрапінгів. Це програмне забезпечення може намагатися автоматично визначити структуру даних сторінки або встановити інтерфейс введення доступний, які пишуть про необхідність вручну коду skrapinh або деяких скриптових функцій, які можуть бути використані для вилучення вмісту і інтерфейсів баз даних і перетворення, Погані дані можуть зберігати локальні бази даних.

Вертикальні агрегаційні платформи: існує кілька компаній, які розробили спеціальну платформу для вертикальної збірки. Ці платформи надають численні і контрольовані «боти» для конкретних галузей без «людини в циклі» (без прямого втручання людини) і без роботи, пов'язаної з конкретним цільовим сайтом. Підготовка включає в себе базу знань для всієї вертикальної специфікації, а потім автоматично створює боти платформи. Платформа надійності вимірює якість одержуваної ними інформації (як правило, кількість полів) і її масштабованість (наскільки швидко вона масштабується до сотень або тисяч сайтів). Ця масштабованість в основному використовується для пунктів призначення [[Long Tail | довгий хвіст]], що загальні агрегатори знаходять складним або занадто трудомістким для збору контенту.

Визнання сторінок семантичної анотації після skrapinhu може розміщувати семантичні метадані або теги і анотації, які можуть використовуватися для пошуку конкретної інформації. Якщо анотації вбудовані в сторінки, роблячи мікроформати, цю процедуру можна розглядати як окремий випадок розбору DOM. В іншому випадку, організовані анотації на семантичному рівні, зберігаються і управляються окремо від мережі, так що

схема може отримувати скребок даних і інструкцій для цього шару сторінок skrapinhom.

3.4. Синтаксичний аналіз або парсинг

Синтаксичний аналіз або парсинг є процесом аналізу ряду символів, або на природній мові, мовах програмування і структури даних відповідно до правил формальної граматики. Термін "синтаксичний аналіз" походить від латинського *pars (orationis)*, це означає, що частинна (мова) мається на увазі.

Термін має дещо різне значення в різних галузях мовознавства та інформатики. Традиційне пропозицію синтаксичного аналізу часто виконується як метод, щоб зрозуміти точний сенс фрази або слова, іноді за допомогою таких пристроїв, як набір діаграм. Це зазвичай підкреслює важливість граматичних підрозділів, таких як предмет і предикат.

У обчислювальної лінгвістиці термін використовується для позначення формального аналізу пропозиції або іншого слова рядка за допомогою комп'ютера в його компоненти, що призводить до синтаксичному дереву із зазначенням їх синтаксичні відносин один до одного, що також може включати в себе семантичну і іншу інформацію ,

Цей термін також використовується в психолінгвістиці при описі розуміння мови. Синтаксичний в цьому контексті посиляється на те, яким чином люди одне речення або фразу (в розмовній мові або текст) «граматичними складові, визначити частини мови, синтаксичних відносин і т.д.» аналізувати. Цей термін особливо поширений при обговоренні того, які лінгвістичні підказки допомагають ораторам інтерпретувати траєкторії дорожніх фраз саду.

У комп'ютерній техніці термін використовується при аналізі

комп'ютерних мов і відноситься до синтаксичного аналізу вхідного коду в його компоненти для полегшення складання компіляторів та інтерпретаторів. Цей термін також може використовуватися для опису поділу або поділу.

Парсер - програмний компонент, який використовує вхідні дані (часто текст) і створює структуру даних - часто це свого роду дерево синтаксису, абстрактне синтаксичне дерево або іншу ієрархічну структуру, яка забезпечує структурне уявлення введення при перевірці правильного синтаксису. Модифікації можуть передувати або слідувати іншим крокам, або вони можуть бути об'єднані в один крок. Парсер часто передує окремий лексичний аналізатор, який генерує токени з послідовності вхідних символів; В якості альтернативи, їх можна комбінувати з разборчіками без сканера. Парсери можуть бути запрограмовані вручну або можуть генеруватися автоматично або напівавтоматично генератором синтаксичного аналізатора. Розбір доповнює шаблонізацію, яка генерує відформатований висновок. Вони можуть бути застосовані до різних доменів, але часто з'являються разом, наприклад, пара `scanf / printf` або генерація зовнішнього коду і генерація вихідного коду компілятора.

Введення парсеру часто є текстом на будь-якому комп'ютерному мовою, але може також бути текстом на природній мові або менш структурованими текстовими даними, і в цьому випадку зазвичай беруться тільки певні частини тексту, а не побудова дерева синтаксичного аналізу. Parsers варіюються від дуже простих функцій, таких як `scanf` до складних програм, таких як інтерфейс компілятора C ++ або аналізатор HTML в веб-браузері. Важливим класом для простого синтаксичного аналізу є використання регулярних виразів, в яких набір регулярних виразів визначає регулярний мову, а механізм регулярних виразів автоматично генерує парсер для цієї мови, який дозволяє розпізнавати і отримувати текст. В інших контекстах замість синтаксичного аналізу

використовуються регулярні вирази, такі як крок лексінга, вихід якого потім використовується парсером.

Використання парсерів варіюється в залежності від введення. У разі мов даних парсер часто вважається функцією читання файлів програми, наприклад: В. читання в HTML або XML-тексті; Ці приклади - мови розмітки. У разі мов програмування парсер є компонентом компілятора або інтерпретатора, який аналізує вихідний код мови комп'ютерного програмування для створення форми внутрішнього представлення; Парсер є важливим кроком в інтерфейсі компілятора. Мови програмування, як правило, вказуються в термінах детермінованою контекстно-вільної граматики, оскільки вони можуть писати швидкі і ефективні парсери. Для компіляторів сам аналіз може бути виконаний за один прохід або кілька проходів - см. Однопрохідний компілятор і багатопрохідний компілятор.

Передбачувані недоліки однопрохідного компілятора можуть бути в значній мірі подолані шляхом додавання виправлень, коли під час переадресації надається кодове зміщення, а виправлення завершуються тому, коли виявлений поточний сегмент програми. Прикладом, в якому такий механізм відновлення був би корисний, була б наступна інструкція GOTO, в якій мета GOTO невідома до завершення сегмента програми. У цьому випадку застосування корекції буде відкладено до тих пір, поки не буде виявлена мета GOTO. І навпаки, зворотний GOTO не вимагає корекції, тому що місце розташування вже відомо.

Контекстно-вільні граматики обмежені тим, що вони можуть висловлювати все мовні вимоги. Неформальна причина, по якій пам'ять такого мови обмежена. Граматика не може запам'ятати існування конструкції по будь-якій довжині введення; Це необхідно для мови, на якому, наприклад, має бути оголошено ім'я до його посилення. Однак більш сильні граматики, які

можуть висловити це обмеження, не можуть бути ефективно проаналізовані. Тому загальною стратегією є створення розслабленого синтаксичного аналізу для контекстно-вільної граматики, яка приймає надмножество бажаних мовних конструкцій (т. Е. Приймає деякі неприпустимі конструкції); пізніше небажані конструкції можуть бути відфільтровані на етапі семантичного аналізу (контекстний аналіз).

3.5. Мова розмітки сторінок HTML

Найважливіший спосіб зберігання та передачі документів в Інтернеті. Дані у форматі HTML схожі на звичайні текстові файли, за винятком того, що деякі символи в них (так звані теги) трактуються як розмітка. Розмітка забезпечує документ певними тегами, структурою, розділами, розділами, параграфами, списками, малюнками, таблицями, заголовками, індексами, вмістом тощо. У кожному блоці можна змінити шрифт, розмір символу та колір тексту, курсивом виділити текст та / або форматувати його жирним шрифтом. Основна функція - це можливість використовувати HTML-посилання, які дозволяють посилання та посилання з веб-сторінки на інші документи, як локальні (поточні сервери), так і сервери у віддалених регіонах світу. Мова HTML також забезпечує введення зображенням зображення, звуку, відео тощо. HTML-документ відображається через браузер (наприклад, Internet Explorer, Netscape тощо).

Недосконалість HTML зробила мережу сумішшю тексту, повного різних елементів і часто використовуваних тегів, що називаються Pro Forma, і безглуздим.

XML пропонує величезний потенціал для оптимізації гіпертексту. Наприклад, для створення посилання в HTML використовується елемент A, а

за допомогою XML ви можете не просто використовувати посилання, наприклад, посилання. В. Створіть двонаправлену посилання.

Перспектива XML полягає в тому, що вона описує інші мови розмітки, такі як JavaScript, які використовуються у HTML-документах.

XML розроблений, щоб спростити та полегшити використання SGML, зберігаючи при цьому великий потенціал для створення, розповсюдження та публікації веб-документів у мережі.

Незважаючи на те, що XML дуже молодий (W3C схвалив специфікацію "Extensible Markup Language (XML) 1.0" на початку лютого 1998 року), окремі компоненти мови все ще близькі до завершення, сьогодні існує нова мова на базі XML Web сервери, які використовують цю технологію для організації інформації, що зберігається на них.

Мова розмітки документа - це набір спеціальних інструкцій, що називаються тегамі для формування документів у структурі та визначення зв'язків між різними елементами цієї структури. Метки мови або, як їх іноді називають, контрольні точки, ці документи якимось кодуєть позицію в основному зміст документа та служать інструкціями для браузера.

Вся краса XML можна зрозуміти лише порівнюючи її з HTML. Форматований в RFC 1866 в 1995 році, HTML є найпопулярнішою мовою розмітки в світі. Звичайно, термін "розмітка" по відношенню до документа означає все, що не відноситься до його вмісту.

У перші дні свого розвитку HTML був використаний як інструмент масштабованого форматування документів, які могли б використовуватися для обміну інформацією практично з будь-якої платформи. В основі HTML є дуже просте поняття: ви визначаєте просту мову, яка описує структуру документа та чекає, поки компанії розроблять програмне забезпечення для подання наступних документів користувачам у різних середовищах. За

допомогою HTML ви можете створювати матеріали, які дозволяють презентації у будь-якому візуальному чи аудіо форматі.

Однак поступово стало зрозуміло, що, незважаючи на свою простоту, ця ідея суперечить загальноприйнятій практиці видавничих систем. Традиційний механізм створення публікацій передбачає, що графічний дизайнер і будівельник повинні враховувати специфічні особливості середовища презентації, включаючи розмір листа, якість друку, колірну палітру тощо. За допомогою цього методу переключився на більш простий, в якому автор несе відповідальність тільки за зміст і логічну структуру документа, представлення зміщення відповідальності за користувацькі програми виявило, що важко.

У файлі HTML у вихідній формі теги форматування змішуються з простим текстом. Звичайно, головна особливість розмітки HTML - це можливість вставляти посилання на зовнішні документи або на внутрішні розділи того ж документа.

HTML розквітає не тільки як індивідуальну мову розмітки, але і як посередницьке програмне забезпечення. Через дешевизну та популярності веб-браузери - чудові клієнти; Через HTML вони можуть спілкуватися з різними серверами.

3.6. Оболонка PyQt

PyQt - зв'язування Python міжплатформного інструментарію GUI Qt, який реалізується як плагін Python. PyQt - це безкоштовне програмне забезпечення, розроблене британською компанією Riverbank Computing. Він доступний в умовах, подібних до версій Qt старше 4,5; Це означає різноманітні ліцензії, включаючи GNU General Public License (GPL) та комерційну ліцензію, але не GNU Lesser General Public License (LGPL). PyQt підтримує

Microsoft Windows, а також декілька варіантів UNIX, включаючи Linux і MacOS (або Дарвін).

PyQt реалізує близько 440 класів і понад 6000 функцій і методів, у тому числі:

- значна кількість віджетів графічного інтерфейсу;
- Класи для доступу до баз даних SQL (ODBC, MySQL, PostgreSQL, Oracle, SQLite);
- QScintilla, віджет багатого текстового редактора на основі Scintilla;
- Віджети, що відстежують дані, які автоматично заповнюються з бази даних;
- XML-аналізатор;
- підтримка SVG;
- Класи для вбудовування елементів керування ActiveX на Windows (лише для комерційної версії).

Щоб автоматично генерувати ці прив'язки, Філ Томпсон розробив інструмент SIP, який також використовується в інших проектах.

У серпні 2009 року Nokia випустила колишній власник інструментарію Qt, PySide і пропонує аналогічні функції, але під LGPL, після угоди з Рівербанк Computing про внесення змін до умов ліцензії для включення в LGPL не вдалося в якості альтернативної ліцензії.

PyQt - це один з найпопулярніших прив'язків Python для платформи C++ для платформи Qt. PyQt був розроблений компанією Riverbank Computing Limited. Сам Qt розробляється як частина проекту Qt. PyQt пропонує прив'язки для Qt 4 і Qt 5. PyQt поширюється під вибором ліцензій: GPL версії 3 або комерційної ліцензії.

PyQt доступний у двох виданнях: PyQt4, який будується проти Qt 4.x і

5.x і PyQt5, які будуються лише з 5.x. Обидві версії можуть бути створені для Python 2 і 3. PyQt містить понад 620 класів, що охоплюють графічні інтерфейси користувача, обробку XML, мережеві зв'язки, бази даних SQL, веб-переглядач та інші технології, доступні в Qt.

Остання версія PyQt - v5.10.1. Він повністю підтримує Qt 5.10.1.

PyQt4 працює на Windows, Linux, Mac OS X і на різних платформах UNIX. PyQt5 також працює на Android і iOS.

PyQt об'єднує крос-платформну платформу програм Qt C ++ і крос-платформну інтерпретовану мову Python.

Qt - це більше, ніж інструментарій для графічного інтерфейсу. Вона містить абстракції мережевих розеток, потоків, Unicode, регулярних виразів. PYQT об'єднує платформу крос-платформної програми Qt C ++ і крос-платформну інтерпретовану мову Python.

Qt - це більше, ніж інструментарій для графічного інтерфейсу. Вона включає в себе абстракцію мережевих сокетів, потоки, Unicode, регулярних виразів, баз даних SQL, SVG, OpenGL, XML, повнофункціональний веб-браузері, система допомоги, мультимедійна основа і великої колекції GUI віджетів. Qt класів використовують механізм сигнал / слот для зв'язку між об'єктами типу-безпечної, але слабо пов'язані, що робить його легко створювати повторно використовувані компоненти програмного забезпечення. Qt також включає в себе Qt Designer, графічного дизайнера інтерфейсу користувача. PyQt може генерувати код Python з Qt Designer. Також можна додати нові елементи керування GUI в Python до Qt Designer.

Python - це проста, але потужна об'єктно-орієнтована мова. Його простота полегшує навчання, але її потужність означає, що великі і складні програми можуть бути створені. Його інтерпретована природа означає, що програмісти Python дуже продуктивні, оскільки не існує циклу розробки

редагування / компіляції / посилання / запуску. Велика частина влади Пайтона виходить з його обширного набору модулів розширення, які надають широкий спектр функцій, включаючи HTTP-сервера, XML парсер, доступ до баз даних, засоби стиснення даних і, звичайно ж, графічні інтерфейси користувача. Модулі розширення зазвичай реалізуються в Python, C або C++. За допомогою таких інструментів, як SIP, відносно легко створити модуль розширення, який інкапсулює існуючу бібліотеку C або C++. Таким чином, Python може стати клей для створення нових програм із встановлених бібліотек.

PyQt поєднує в собі всі переваги Qt та Python. Програміст має всю міць Qt, але може використовувати її з простотою баз даних Python.SQL, SVG, OpenGL, XML, повнофункціональний веб-браузер, систему допомоги, мультимедійне основи і великої колекції віджетів GUI. Qt класів використовують механізм сигнал / слот для зв'язку між об'єктами типу-безпечної, але слабо пов'язані, що робить його легко створювати повторно використовувані компоненти програмного забезпечення.

Qt також включає в себе Qt Designer, графічного дизайнера інтерфейсу користувача. PyQt може генерувати код Python з Qt Designer. Також можна додати нові елементи керування GUI в Python до Qt Designer. Python - це проста, але потужна об'єктно-орієнтована мова. Його простота полегшує навчання, але її потужність означає, що великі і складні програми можуть бути створені. Його інтерпретована природа означає, що програмісти Python дуже продуктивні, оскільки не існує циклу розробки редагування / компіляції / посилання / запуску.

Велика частина влади Пайтона виходить з його обширного набору модулів розширення, які надають широкий спектр функцій, включаючи HTTP-сервера, XML парсер, доступ до баз даних, засоби стиснення даних і, звичайно ж, графічні інтерфейси користувача. Модулі розширення зазвичай

реалізуються в Python, C або C ++. За допомогою таких інструментів, як SIP, відносно легко створити модуль розширення, який інкапсулює існуючу бібліотеку C або C ++. Таким чином, Python може стати клей для створення нових програм із встановлених бібліотек. PyQt поєднує в собі всі переваги Qt та Python. Програміст має всю потужність Qt, але може використовувати його з простотою Python.

3.7. Система керування базами даних MySQL

Система керування базами даних - це комп'ютерна програма або набір програм, що надає користувачам можливість створювати, зберігати, оновлювати, шукати інформацію та бази даних контролю доступу. Система керування базами даних (СУБД) поєднує в собі мовні та програмні засоби, призначені для створення, керування та обміну даними з багатьма користувачами.

Сервер MySQL був обраний як сервер баз даних. Система управління базами даних MySQL - це найбільш поширені системи, які можна використовувати для роботи з реляційними базами даних. Спочатку він використовується для створення динамічних веб-сторінок, оскільки він має відмінну підтримку різних мов програмування. Система MySQL вважається гарним рішенням для малих і середніх застосунків. Серверні серійні коди складаються на багатьох платформах. Найбільш повні функції сервера знаходяться в системах UNIX, які мають багатопоточну підтримку, що збільшує загальну продуктивність системи.

Мова SQL - декларативна база даних взаємодії з користувачами з базами даних, яка використовується для генерації запитів, оновлення та управління реляційними базами даних, створення схеми бази даних та її модифікацій,

систем контролю доступу до бази даних. Сам SQL не є ні системою керування базами даних, ні окремим програмним продуктом. Оскільки SQL не є мовою програмування в сенсі C або Pascal, вона може створювати інтерактивні запити або, якщо вони інтегровані в додатки, служать інструкціями для управління даними. Стандарт SQL також включає функціональність для визначення змін, перевірки та конфіденційності.

Мова SQL є діалогічною мовою програмування для запитів та зміни бази даних, а також для адміністрування баз даних. Багато баз даних підтримують SQL із стандартними розширеннями мовлення. Серент SQL - це мова команди, яка дозволяє виконувати пошук, вставка, оновлення та видалення даних за допомогою системи управління та адміністративних функцій.

Цей сервер бази даних має великий вибір баз даних кластеризації та віддзеркалення, що є важливим для побудови системи з високою толерантністю до помилок та швидкою продуктивністю. Система управління базами даних (СУБД) MySQL - розробка шведської компанії MySQL AB. База MySQL являє собою програмне забезпечення cookie з відкритим вихідним кодом, поширене під GNU General Public License (GPL) та комерційні ліцензії для ситуацій, які не охоплюються GPL. MySQL підтримує реляційні дані, що означає, що це реляційна база даних.

Станом на версію 5.0 база даних MySQL практично повністю відповідає структурованій мові стандартної побудови та SQL-запитів, отже, вона сумісна з іншими базами даних.

Розглянемо основні переваги MySQL:

- Висока якість - MySQL характеризується стабільною роботою;
- Спільно з Oracle, MySQL вважається однією з найшвидших в світі СУБД;
- Open Source доступний для перегляду та оновлення, що дозволяє

постійно вдосконалювати програмний продукт;

- СУБД MySQL, розроблений з мовами C / C ++, протестований на багатьох платформах, включаючи Windows, Linux, FreeBSD, Mac OS X, OS / 2, Solaris та інші;

- MySQL підтримує API (Application Programming Interface) для C, C ++, Eiffel, Java, Perl, PHP, Python, Ruby і Tcl. MySQL можна успішно використовувати для створення веб-сторінок за допомогою Perl, PHP та Java, а також для програми, створеної за допомогою Delphi, C ++ Builder або .NET платформи;

- База даних MySQL пропонує широкий спектр типів таблиць, у тому числі сторонніх розробників, що оптимізує проблему продуктивності та функціональності;

- Локалізація в MySQL правильна.

Також у версії MySQL 4.1 існують такі важливі нововведення, як повна підтримка вкладеного запиту та підтримка транзакцій. І в MySQL 5.0 з'явилися наступні основні механізми:

- Збережені процедури та функції, які поєднують всю послідовність вимог;

- Тригер, д. Н. збережені процедури пов'язані з подією зміни таблиці;

- Перегляди - зразки даних, які можуть бути представлені як повноцінні реальні таблиці бази даних;

- Курсорів, які дозволяють циклу відображати кожен рядок результуючої таблиці запитів.

Інформаційна схема, тобто портативний набір переглядів системної таблиці, в якій зберігається різноманітна внутрішня інформація.

3.8. Засоби обробки інформації

У сучасному програмному забезпеченні існує безліч різних програмних засобів для обробки інформації, написаної на різних мовах програмування, на основі вищенаведених методів. Варіативність ПП пов'язана з особливостями кожної галузі, в якій здійснюється обробка. Наприклад, при обробці графічних зображень широко застосовуються методи розпізнавання образів, криптографічні методи. На основі перетворення Фур'є і так далі.

Не кожна компанія може дозволити собі призначити розробника програми ПП, яка ефективно управляє інформацією, яка конкретно стосується розміру цієї компанії. Такий підхід навіть бажаний, оскільки автоматизовані системи обробки базуються на визначеній базі даних, структура якої може різнитися в різних компаніях, не кажучи вже про різні галузі. Одним з найпоширеніших інструментів обробки інформації є пакет Microsoft Office, який встановлюється практично на кожному комп'ютері. Функціональність досить широка, але примітивна, якщо користувач не може працювати безпосередньо в програмному середовищі, для якого призначений офіс. Серед інструментів, доступних для широкого кола споживачів, є організація баз даних або виконання запитів та пошуку інформації, фільтрування інформації, графічне подання тощо.

Сьогодні бухгалтер працює рідко без програми бухгалтерського обліку 1С та компанії 1С, яка служить для зберігання та обробки бухгалтерської інформації. Програма автоматично створює звіти на основі введених даних про паперові документи. Для обробки графічних зображень (відсканованих) використовується Fine Reader, відмінний приклад розпізнавання інформації. Звукові редактори використовуються для роботи з аудіофайлами. Наприклад, Sound Forge. Для конкретного та детального

аналізу статистичних даних ми використовуємо пакет SPSS, розроблений спеціально для обробки даних за статистичними методами (тести гіпотез, графічне представлення тощо).

В даний час SCADA система спостереження та збору даних (SCADA) все частіше використовується в результаті глобального розподілу комп'ютерних систем у галузі автоматизації промислових процесів. SCADA - це лише один з компонентів автоматизованих систем управління, які в даний час є складним набором програмних та апаратних засобів. Переважна більшість автоматизованих систем керування базуються на промислових контролях, які є основним засобом збирання, обробки інформації, контролю технологічних параметрів, сигналізації, захисту та замків (нижчий рівень системи). Інформація, оброблена контролерами, передається комп'ютерним системам, що представляють робочу станцію технічного працівника оператора, де дані процесу та презентації додатково обробляються для оператора інтуїтивно (верхній рівень системи управління).

Системи SCADA в ієрархії апаратної автоматизації апаратної промислової автоматизації знаходяться на верхньому рівні. Якщо ми спробуємо коротко описати основні функції, то можна сказати, що система SCADA збирає інформацію про технологічний процес, інтерфейси з оператором, зберігає історію процесу та керує процесом у тому обсязі, в якому він знаходиться. Система SCADA - моніторинг та збір даних. Це група засобів контролю та моніторингу та спосіб взаємодії з технологічним об'єктом. На сьогоднішній день цей термін розуміється як сукупність програмних та апаратних засобів для виконання операторських завдань.

При створенні інформаційної системи велика кількість локальних підсистем, які часто мають різні інтерфейси програми, об'єднуються для створення функціонального вузла. Це значно ускладнює завдання

збалансування подібних підсистем та знижує продуктивність системи в цілому. Тому доцільно підключати локальні вузли функцій до однієї і тієї ж програмної платформи. SCADA - система реалізує цей підхід, тому його застосування зараз широко розповсюджене та актуальне. Апаратна частина також розроблена безпосередньо для програмного пакета, що дозволяє вам створити інформаційну систему більш економічно і в мінімальний проміжок часу.

3.9. Технології використані при програмній реалізації

Rich Standard Library є однією з привабливих бібліотек Python. В ній є інструменти для роботи з багатьма мережевими протоколами та форматами Інтернету, такими як модулі для написання HTTP-серверів і клієнтів для аналізу та створення електронної пошти, для роботи з XML тощо. Набір модулів для роботи з операційною системою, ви можете писати крос-платформні програми. Є модулі для роботи з регулярними виразами, кодування тексту, мультимедійні формати, криптографічні протоколи, архіви, серіалізація даних, підтримка модульних тестів та інші.

Якщо ви виходите з інтерпретатора Python і входите в нього знову, визначення, які ви зробили (функції і змінні) губляться. Отже, якщо ви хочете написати якусь довшу програму, вам краще використовувати текстовий редактор для підготовки введення для інтерпретатора, і передати код з цього файлу в якості вхідних даних. Це називається створенням скриптів. Коли ваша програма стає довшою, ви можете захотіти розділити її на кілька файлів для більш легкого супроводу. Ви також можете захотіти використовувати зручну написану вами функцію в декількох програмах без копіювання її визначення в кожену програму.

Для підтримки цього в Python є спосіб помістити визначення в файл і використовувати їх в скриптах або в інтерактивному режимі інтерпретатора. Такий файл називається модулем; визначення з модуля можуть бути імпортовані в інші модулі або в головний модуль (колекція змінних, до яких ви маєте доступ в виконуваному скрипті на верхньому рівні і в режимі калькулятора).

Модуль - це файл, який містить визначення і оператори Python. Файл є ім'ям модуля з доданим суфіксом `.py`. У модулі ім'я модуля (як рядок) є як значення глобальної змінної `__name__`.

Поряд з визначеннями функцій модулі можуть містити виконувані оператори. Ці оператори призначені для ініціалізації модуля. Вони виконуються лише один раз, коли в операторі імпорту зустрічається ім'я модуля. (Вони також запускаються, якщо файл виконується як скрипт.)

Кожен модуль має його власну приватну таблицю імен, яка використовується як глобальна всіма функціями, визначеними в модулі. З іншого боку, якщо ви знаєте, що робите, то можете використовувати глобальну змінну модуля з такою ж нотацією, яка використовується для посилання на його функцію, `modname.itemname`.

Модулі можуть імпортувати інші модулі. Прийнято, але не є необхідним, розміщувати всі оператори `import` на початку модуля (це стосується і скрипта). Ім'я імпортованого модуля поміщається в глобальну таблицю імен імпортує модуля.

Python має велику стандартну бібліотеку, зазвичай згадується як один з найсильніших в мові Python, що забезпечують інструменти підходять для багатьох завдань. Це навмисне і був описаний як "батареї, включених" філософії Python. Для інтернет-додатків перед, велика кількість стандартних форматів і протоколів (наприклад, MIME і HTTP) підтримуються. Модулі для

створення графічних користувацьких інтерфейсів, підключення до реляційних баз даних, генератори псевдовипадкових чисел, арифметиці з довільною точністю знаків після коми, маніпулювання регулярних виразів, і робить модульне тестування, також включені.

Стандартна бібліотека не є необхідним для запуску Python або вставляти Python в додатку. Blender 2.49, наприклад, опускає більшу частину стандартної бібліотеки.

Станом на січень 2016 року, Пітон індекс Пакет, офіційний репозиторій стороннього програмного забезпечення для Python, містить більш ніж 72000 пакетів, що пропонують широкий спектр функціональних можливостей, в тому числі:

- графічні інтерфейси, веб-рамки, мультимедіа, бази даних, мережі і комунікації;
- тест рамки, автоматизація та веб-зішкріб, інструменти документації, системне адміністрування;
- наукових обчислень, обробки тексту, обробка зображень.

Висновки до розділу 3

Обрана конфігурація дозволила забезпечити високу якість, гнучкість та швидкий розвиток. Серверу додатків не довелося керувати багатьма залежностями, скоротити час, необхідний для налаштування та налаштування програм і служб, а також зосередити увагу на розробці програмного продукту.

4. ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

Для покращення роботи програмного забезпечення розроблені додаткові функції та типи даних, що забезпечує більш високий рівень продуктивності, надійність системи та легкість сприйняття коду.

Розроблений програмний засіб складається з наступних основних частин:

- База даних;
- Програмна частина;
- Клієнтська частина.

4.1. Структурна модель

У проекті використовувалася технологія реляційного картографування об'єктів (ORM), яка пов'язувала базу даних з класами, що використовуються в програмі.

Для реалізації структури проекту було створено п'ять класів з вертикальною залежністю, тобто кожен наступний клас міститься в попередньому:

1. Клас Search — проект являється основним об'єктом для пошуку. Складається з ключових слів та вибору бібліотечного джерела. Пошук є об'єктом пошуку.

2. Клас Keyword — етапи проекту. В який заповнюється ключовими словами та результатом.

3. Клас Search_Source — являє собою запитів до вибору джерела, та саме вибору пошукового запиту по назві та ключовим словам посилання та мова джерела.

4. Клас Users — Задається короткі відомості про користувача та надає можливість зареєстрованим користувачам користуватися програмою.

5. Клас Library_source — являє собою збірку джерел, якіх задається назва та короткий опис посилання та мова джерела.

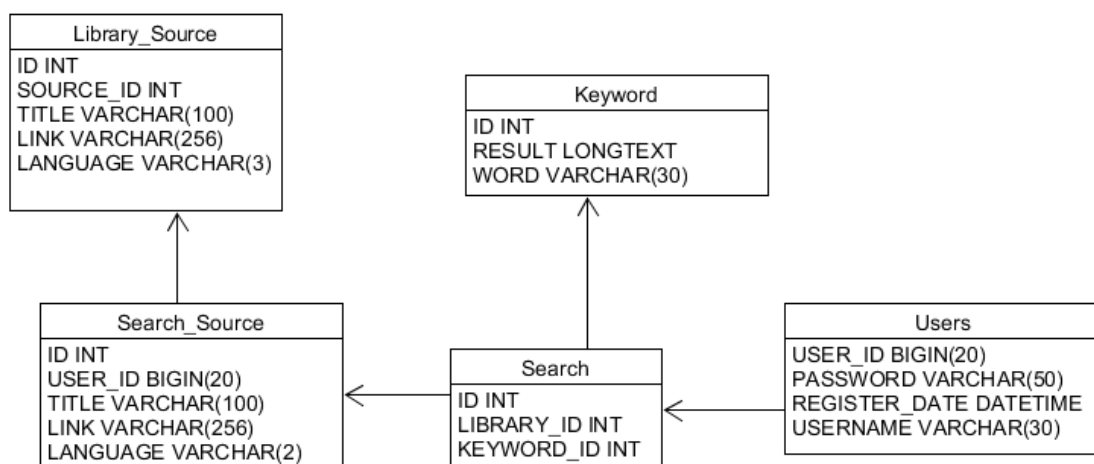


Рисунок 4.1 — Модель елементів проекту

4.2. Діаграма прицендентів

Діаграма прецендентів є графом, що складається з множини акторів, прецендентів (варіантів використання) обмежених границею системи (прямокутник), асоціацій між акторами та прецендентами, відношень серед прецендентів, та відношень узагальнення між акторами. Діаграми прецендентів відображають елементи моделі варіантів використання.

Суть даної діаграми полягає в наступному: проєктована система

представляється у вигляді безлічі сутностей чи акторів, що взаємодіють із системою за допомогою так званих варіантів використання. Варіант використання (*use case*) використовують для описання послуг, які система надає актору. Іншими словами, кожен варіант використання визначає деякий набір дій, який виконує система при діалозі з актором. При цьому нічого не говориться про те, яким чином буде реалізована взаємодія акторів із системою.

Діаграми Прецедентів відіграють важливу роль не тільки у комунікації між збирачами вимог до проекту і потенційними користувачами. Діаграми Прецедентів дописані бізнес логікою і детальними специфікаціями прецедентів, як джерельна інформація, успішно використовують учасники розробки проекту на всіх його фазах (зародження, дизайн, програмування, тестування, документування..). Добре продумані і завершені специфікації прецедентів легко перевтілюються у Тестові Випадки.

Елементи взаємодії Діаграми Прецедентів:

- Використовує – користувач виконує дію;
- Включає – один прецедент використовує іншого;

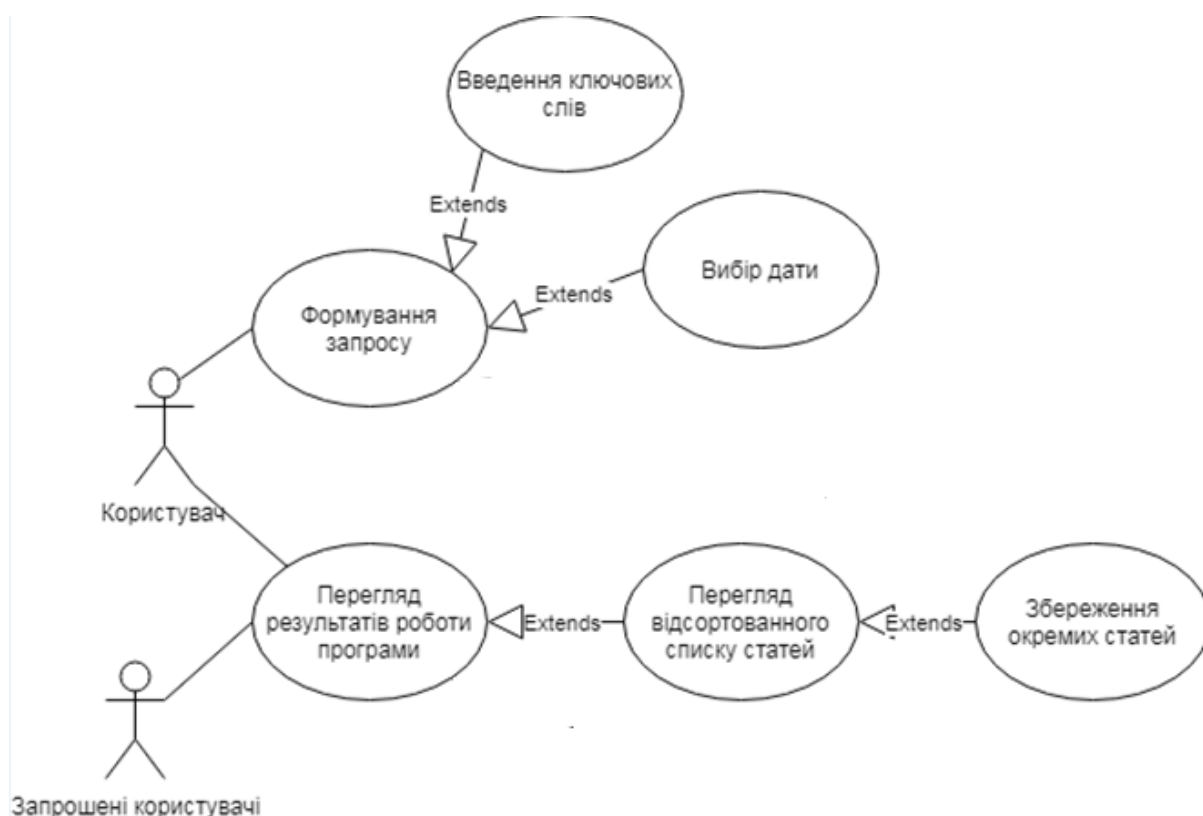


Рисунок 4.2 — Діаграма прицендентів

4.3. Фреймворк в Qt Designer

За допомогою Qt Designer ви можете створювати графічні інтерфейси користувача з різними інструментами. Існує Панель інструментів "Віджет панелі", в якій доступні елементи інтерфейсу - віджети, такі як розкривний список ComboBox, поле введення LineEdit, кнопка PushButton та багато інших. Кожен віджет має власні властивості, визначені відповідним класом бібліотеки Qt. Властивості віджету можна змінити за допомогою "Редактора властивості". Для кожного класу властивостей віджету є спеціальний редактор. Характерною особливістю Qt Designer є підтримка візуальної обробки сигналів і слотів. Наприклад, ви можете призначити сигнал, який генерується,

перемикаючи статус віджета CheckBox в слот, який відповідає за наявність іншого віджета.

Poisk.py - це файл макета, який зберігає розмітку того, що наразі на екрані. Цей файл створює користувацький інтерфейс: поля введення, елементи управління, такі як кнопка, CheckBox, TextView, SurfaceView тощо.

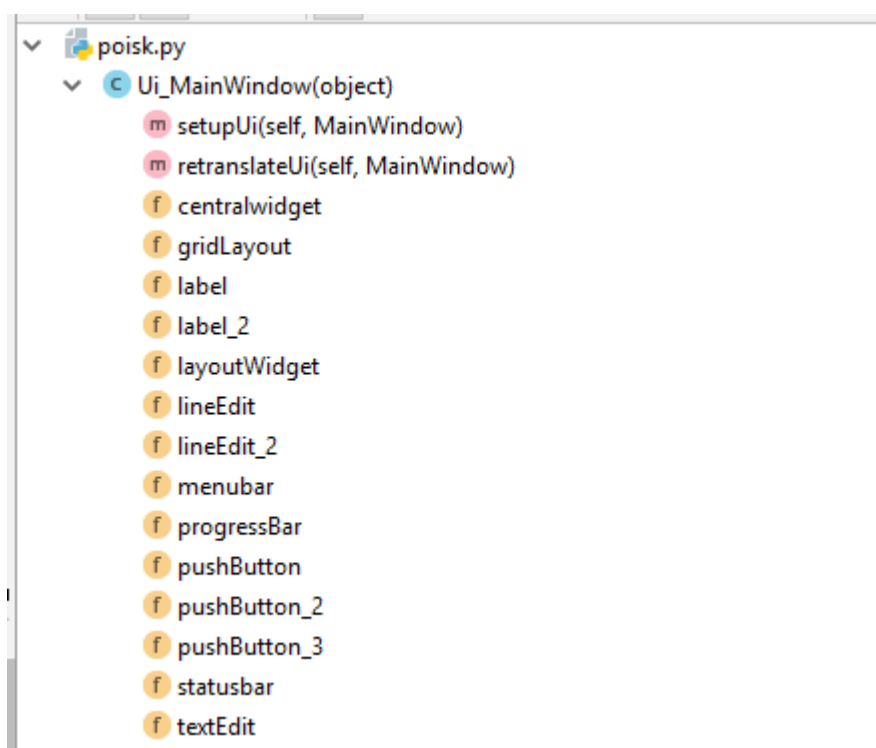


Рисунок 4.3 — Методи класу MainWindow

4.4. Обробка отриманої інформації

У обчислювальній техніці обробка інформації широко відноситься до використання алгоритмів перетворення даних - визначальної активності комп'ютерів; Дійсно, широка професійна організація обчислень відома як Міжнародна федерація обробки інформації (IFIP). Це по суті є синонімом термінів обробки даних або обчислень, хоча з більш загальним коннотацією.

```

savess = doc.split("\n")
for resf in savess:
    resf = resf.strip()
    if ((len(resf) > 50) and (resf.startswith('&') == False) and (resf.startswith('>') == False) and (
        resf.startswith('*') == False) and (resf.startswith('\n') == False) and (
        resf.startswith('<') == False) and (resf.startswith('(') == False) and (
        resf.startswith('#') == False) and (
        resf.endswith('.') or resf.endswith('?') or resf.endswith('!'') or resf.endswith(';'))):
        sm = sm + resf + "\n\n"
if (len(sm) > 500):
    self.texts.append(sm)

```

Рисунок 4.4 — Реалізація обробки тексту для видачі

Системи обробки даних — комплекс взаємопов'язаних методів і засобів збору та обробки даних, необхідних для організації управління об'єктами та вирішення проблем автоматизованим способом. Системи обробки даних ґрунтуються на застосуванні електронних обчислювальних машин і інших сучасних засобів інформаційної техніки, тому їх також називають автоматизованими системами обробки даних. Без електронної обчислювальної машини побудова системи обробки даних можливо тільки на невеликих об'єктах.

Система обробки даних може включати в себе деяку комбінацію:

- Перетворення перетворення даних в іншу форму або мову;
- Перевірка - Переконайтеся, що надані дані є "чистими, правильними та корисними";
- Сортування - "упорядкування елементів у певній послідовності та / або в різних наборах";
- Підведення підсумків - скорочення детальних даних до основних моментів;
- Агрегація - об'єднання декількох частин даних;
- Аналіз - "збір, організація, аналіз, інтерпретація та представлення даних";
- Звітність - детальна інформація про реквізити або зведені дані або

обчислювану інформацію.

Висновки до розділу 5

Розроблена структура дозволила повною мірою використовувати шаблон архітектури, реалізований на основі Qt Designer, що призвело до слабкого зв'язку програмних модулів. За допомогою цього підключення ви можете змінювати вже написані модулі без втрати попередньої функції, за умови, що ми не змінюємо джерела для цих модулів.

5. РОБОТА КОРИСТУВАЧА З ПРОГРАМНОЮ СИСТЕМОЮ

Для того щоб почати користуватись розробленим програмним забезпеченням необхідно ознайомитись з наступними пунктами.

5.1. Рекомендації до апаратного і програмного забезпечення

Таблиця 5.1 — Рекомендоване програмне та апаратне забезпечення

Операційна система	Windows Android OS X Linux
Наявність інструментів	Python 3.6 мая бути встановлений Доступ до мережі Інтернет
Роздільна здатність	1024x768 або вище

При недотриманні рекомендацій сервісом можна буде користуватися, проте деякі інформаційні вікна не будуть доступні та відображення системи буде спотворено.

Так як розроблена система не працює в режимі онлайн, програма потребує інсталяції.

Основна розробка програмного продукту поділялася на 2 частини: побудова інтерфейсу та організація коректної роботи обчислювальних методів.

5.2. Робота користувача з програмою

У системі передбачено типи користувачів із відповідним функціоналом (рисунок 6.1):

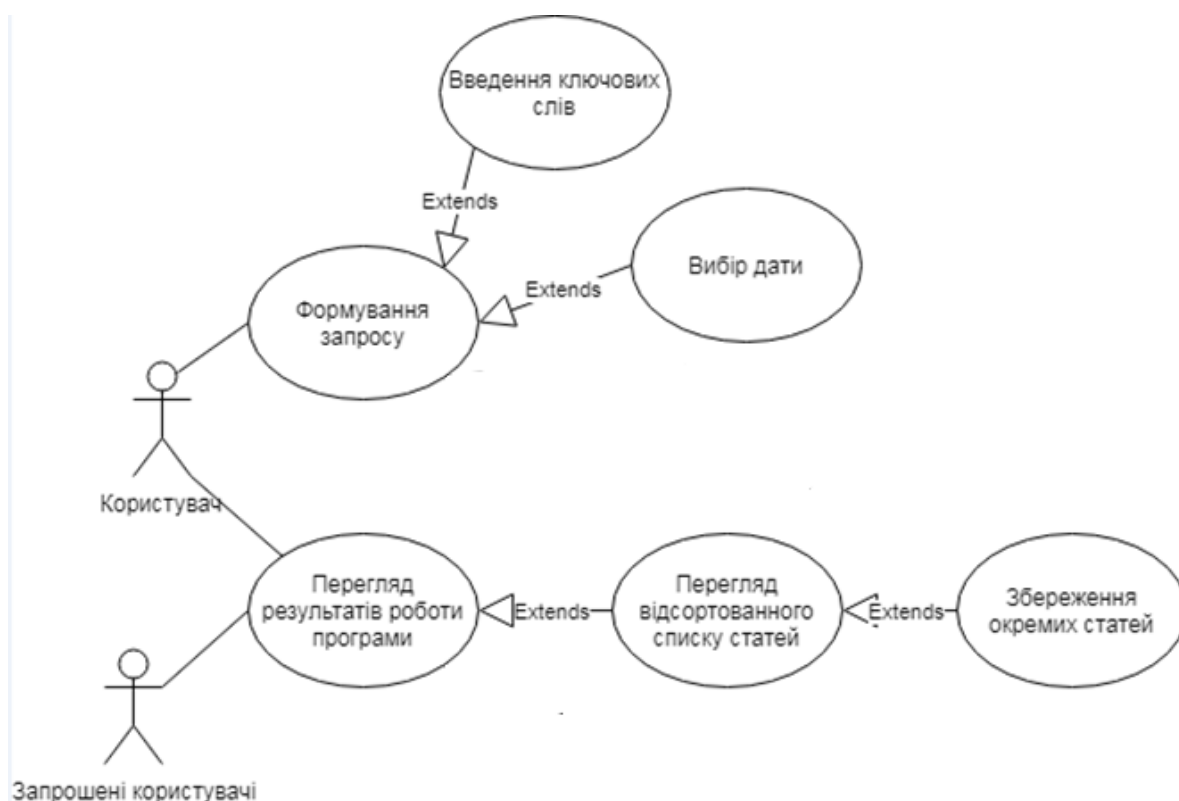


Рисунок 6.1 — Діаграма use case розробленої системи

REST, спеціальний підхід до архітектури мережесих протоколів також використовувався при розробці програмного продукту.

REST (Representational State Transfer) - це підхід до архітектури мережевого протоколу, що забезпечує доступ до інформаційних ресурсів. засновників протоколу HTTP. REST ґрунтується на принципах World Wide Web та, зокрема, на можливостях HTTP. Філдінг розробляв REST паралельно з HTTP 1.1 на основі попереднього протоколу HTTP 1.0.

5.3 Інтерфейс пошуку статей

Дані повинні передаватися у невеликій кількості стандартних форматів (наприклад, HTML, XML, JSON). Протокол REST (наприклад, HTTP) повинен підтримувати кешування, не повинен залежати від мережевого рівня, і не повинен зберігати інформацію про статус між парою запитів / відповідей. Стверджується, що цей підхід дозволяє масштабувати систему і дає йому можливість розвиватися з новими вимогами.

Ключові слова пошуку	Дата виходу статті	Назва автору статті	Кількість стеттей
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="3"/>
<< СТАТТЯ	СТАТТЯ >>	<input type="text" value="0%"/>	НАЙТИ

Рисунок 6.2 — Інтерфейс введення даних для пошуку

Підхід "Antipodes REST" базується на RPC (Remote Procedure Call, RPC). За допомогою підходу RPC можна використовувати невелику кількість мережевих ресурсів з великою кількістю методів і складних протоколів. Під час наближення до REST число методів та складність протоколу суворо обмежені, а це означає, що кількість окремих ресурсів повинна бути великою.

REST - це архітектура розподілених гіпертекстових систем.

Відповідь, отримана службою, є об'єктом JSON, спеціальна позначка, широко використовується в службах REST.

JSON базується на двох структурах:

- Назва / значення пари;
- Порядковий список значень.

Це універсальна структура даних. Теоретично вони підтримують всі сучасні мови програмування в тій чи іншій формі. Оскільки JSON використовується для обміну даними між різними мовами програмування, має

сенс побудувати їх на цих структурах.

Масив - це послідовність значень. Масив починається символом [і закінчується символом]. Значення розділені комами. Значення може бути рядком в подвійних лапках, або числом, або логічними true чи false, або null, або об'єктом, або масивом. Ці структури можуть бути вкладені одна в одну.

Рядок — це послідовність з нуля або більше символів юнікода, обмежена подвійними лапками, з використанням escape-послідовностей, що починаються із зворотної косої риски (backslash). Символи представляються простим рядком.

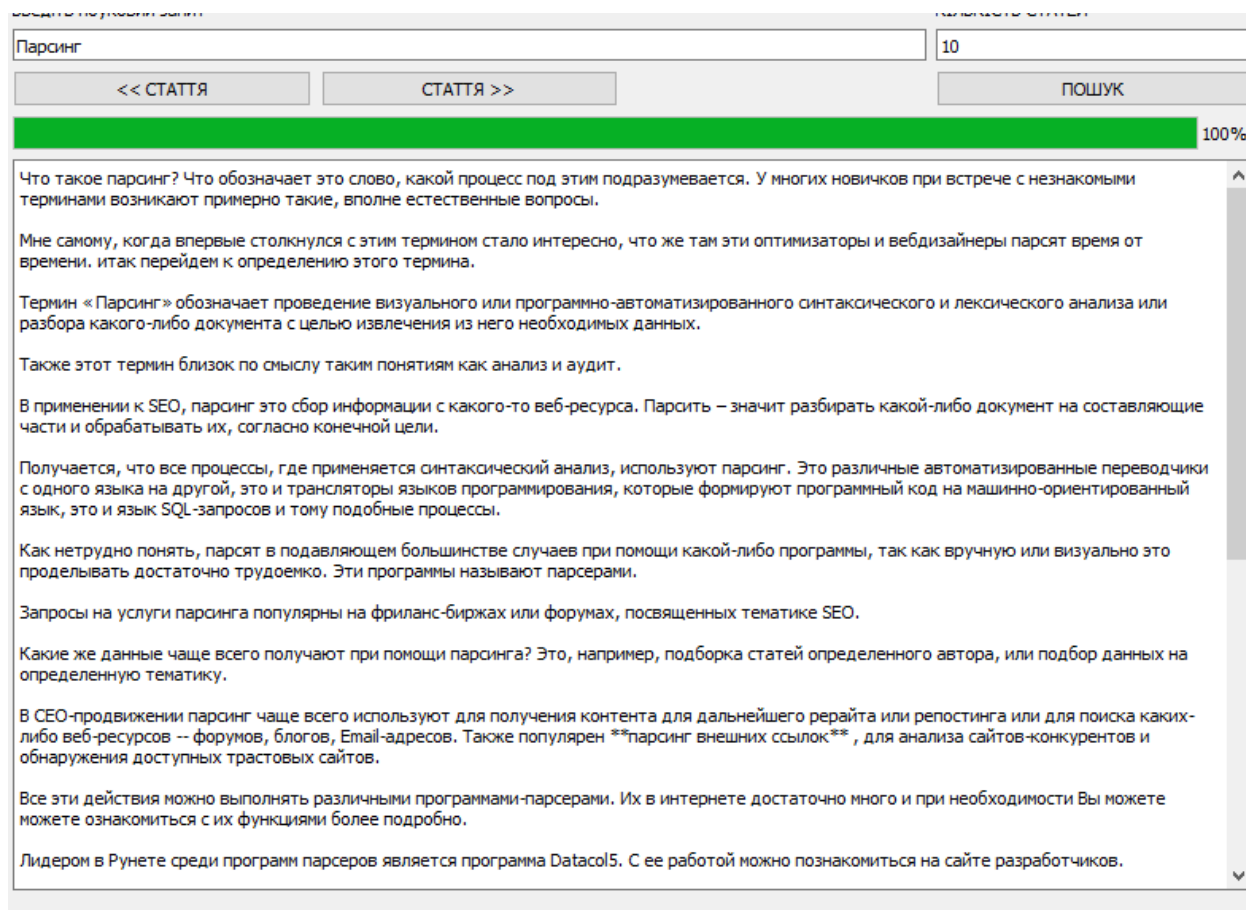


Рисунок 6.3 — Інтерфейс виведення даних пошуку

Висновки до розділу 5

Інтерфейс користувача був розроблений таким чином, щоб бути якомога інтуїтивно зрозумілим і полегшити для різних нових і знайомих користувачів системи працювати з системою.

Запрошеному користувачеві було надано просту навігацію по проекту, переглянувши їх статистику та питання, на які потрібно відповісти, можливість зберігати проекти своїх відповідей та анонімність перед групою організаторів, щоб виключити можливий тиск на їхню частину і виключити можливість перегляду відповідей інших експертів на скорочення. Конформність учасників.

6. СТАРТАП ПРОЕКТ

Розділ має на меті проведення маркетингового аналізу стартап проекту задля визначення принципової можливості його ринкового впровадження та можливих напрямів реалізації цього впровадження. Проведення маркетингового аналізу передбачає виконання нижченаведених кроків.

6.1 Опис ідеї проекту

В межах підпункту слід проаналізувати та подати у вигляді таблиць:

- зміст ідеї (що пропонується);
- можливі напрямки застосування;
- основні вигоди, що може отримати користувач товару;
- чим відрізняється від існуючих аналогів та замінників.

Перші три пункти подаються у вигляді таблиці (таблиця 6.1) і дають цілісне уявлення про зміст ідеї та можливі базові потенційні ринки.

Таблиця 6.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Формування сценаріїв розвитку пошуку інформації методами форсайту	1. Системи формування сценаріїв	1. Автоматизація формування сценаріїв
	2. Системи експертного оцінювання	2. Оптимізація часу експертного оцінювання
	3. Статичний аналіз проектів	3. Не залежить від таргетингової платформи

Аналіз потенційних техніко-економічних переваг ідеї (чим відрізняється

від існуючих аналогів та замінників) порівняно із пропозиціями конкурентів передбачає:

визначення переліку техніко-економічних властивостей та характеристик ідеї;

визначення попереднього кола конкурентів (проектів-конкурентів) або товарів-замінників чи товарів-аналогів, що вже існують на ринку, та проводиться збір інформації щодо значень техніко-економічних показників для ідеї власного проекту та проектів-конкурентів відповідно до визначеного вище переліку;

проводиться порівняльний аналіз показників: для власної ідеї визначаються показники, що мають а) гірші значення (W, слабкі); б) аналогічні (N, нейтральні) значення; в) кращі значення (S, сильні) (таблиця 6.2).

Таблиця 6.2. Визначення сильних, слабких та нейтральних характеристик

№ п/п		(потенційні) товари/концепції конкурентів		
		Мій проект	Системи формування сценаріїв	Системи експертного оцінювання
1	W слабка сторона	Відсутність графічного редактора	Матеріали додають лише власники порталу	Висока ціна
2		Відносно повільний час роботи	Повільний час роботи	Не враховує особливості формування сценаріїв
3	N нейтральна сторона	Можливість задавати валідацію даних	Не обирається тема для анкет	Експерт сам обирає тему анкет

4	S сильна сторона	Побудований сценарій можна представити у вигляді діаграм для більшої наочності	Комбінація внутрішнього та зовнішнього аналізатора	Комплексна система додатків
5		Формування сценарію відбувається автоматично	Формування сценарію відбувається автоматично	Можливо використовувати в різних предметних областях

Таблиця 6.2. (продовження)

6.2 Технологічний аудит ідеї проекту

В межах даного підрозділу необхідно провести аудит технології, за допомогою якої можна реалізувати ідею проекту. Визначення технологічної здійсненності ідеї проекту передбачає аналіз таких складових (таблиця 6.3):

- за якою технологією буде виготовлено товар згідно ідеї проекту;
- чи існують такі технології, чи їх потрібно розробити/доробити;
- чи доступні такі технології авторам проекту.

Таблиця 6.3. Технологічна здійсненність ідеї проекту

№ п/п	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Інтерфейс користувача	PyQt5, HTML, CSS	Наявна	Доступна безкоштовно
2	Розширювана база даних	SQL, MySQL Server	Наявна	Доступна безкоштовно
3	Алгоритм створення анкет	HTML, CSS, jQuery	Наявна	Відсутня

Висновок: проект реалізувати можливо.

Обрана технологія реалізації ідеї проекту: Нативний інтерфейс користувача

За результатами аналізу таблиці робиться висновок щодо можливості технологічної реалізації проекту: так чи ні, а також технологічного шляху, яким це доцільно зробити (з поміж названих технологій обираються такі, що доступні авторам проекту та є наявними на ринку).

6.3 Аналіз ринкових можливостей запуску стартап-проекту

Визначення ринкових можливостей, які можна використати під час ринкового впровадження проекту, та ринкових загроз, які можуть перешкодити реалізації проекту, дозволяє спланувати напрями розвитку проекту із урахуванням стану ринкового середовища, потреб потенційних клієнтів та пропозицій проектів-конкурентів.

Спочатку проводиться аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку (таблиця 6.4).

Таблиця 6.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/ п	Показники стану ринку (найменування)	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн/ум.од	425 грн
3	Динаміка ринку (якісна оцінка)	Зростає
4	Наявність обмежень для входу (вказати характер обмежень)	Немає

5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі (або по ринку), %	50 %

Середня норма рентабельності в галузі (або по ринку) порівнюється із банківським відсотком на вкладення. За умови, що останній є вищим, можливо, має сенс вкласти кошти в інший проект.

За результатами аналізу таблиці робиться висновок щодо того, чи є ринок привабливим для входження за попереднім оцінюванням.

Надалі визначаються потенційні групи клієнтів, їх характеристики, та формується орієнтовний перелік вимог до товару для кожної групи (таблиця 6.5).

Таблиця 6.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
1	Автоматизація агрегації програмного коду	Інформаційні системи прогнозування	Компанії заключають довготривалі договори, а стартапери віддають перевагу пробному терміну	стабільність роботи Невисока ціна Наявність пробного періоду Наявність документації Підтримка необхідних платформ

Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (таблиці 6.6-6.7).

Таблиця 6.6. Фактори загроз

№ п/ п	Фактор	Зміст загрози	Можлива реакція компанії
1	Підходить для нових проєктів	Потребує визначеної структури бази даних	Імпорт схеми бази даних
2	Власний формат зберігання	При необхідності потрібна розробка сервісу преведення до визначеного формату	Додавання можливості автоматизованого експорту в різні типи сховищ, розробка додаткового ПЗ
3	Обмеженість функцій	Інструмент обмежений наявними функціями і не має деяких функцій, які мають конкуренти	Додавання нових функцій за потреби

Таблиця 6.6. (продовження)

Таблиця 6.7. Фактори можливостей

№ п/ п	Фактор	Зміст можливості	Можлива реакція компанії
1	Незалежність від платформи	Можна використовувати як web інтерфейс	Вихід на мобільний ринок, вихід на рівень web додатків
2	Недоліки в існуючих альтернативах	Існуючі альтернативи або працюють повільно, або не є орієнтованими на конкретну предметну область	Модифікація існуючих платформ

Надалі проводиться аналіз пропозиції: визначаються загальні риси конкуренції на ринку. Аналіз пропозиції необхідно виконати аналізуючи існуючі види конкуренцій.

Пропозиції повинні відповідати на питання “Як просувати продукт”.

Аналіз пропозицій зображено на таблиці 6.8.

Таблиця 6.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1. Вказати тип конкуренції - монополія/олігополія/ монополістична/чиста	чиста	Прямі договори з стартапами, презентація продукту на виставках
2. За рівнем конкурентної боротьби - локальний/національний /...	національний	Публікація статей на міжнародних сайтах
3. За галузевою ознакою - міжгалузева/ внутрішньогалузева	внутрішньогалузева	Розвивати напрямки, нерозвинуті конкурентами
4. Конкуренція за видами товарів: - товарно-родова - товарно-видова - між бажаннями	товарно-видова	Розповідати про свої переваги перед конкурентом у цій галузі
5. За характером конкурентних переваг - цінова / нецінова	нецінова	Надання функцій, які не надають конкуренти

6. За інтенсивністю - марочна/не марочна	марочна	Надання функцій, які не надають конкуренти
---	---------	--

Після аналізу конкуренції проводиться більш детальний аналіз умов конкуренції в галузі (таблиця 6.9).

Таблиця 6.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники (Архітектори БД)	Клієнти (Розробники)	Товари-замінники
	ReSharper SonarQube		Мінімізація витрат часу постачальників	Контроль якості	Лояльність споживачів
Висновки :	Визначити інтенсивність конкурентної боротьби з боку прямих конкурентів	Є можливості виходу на ринок, оскільки існуючі рішення не надають потрібних переваг	Постачальники підлаштовуються під ринок	Клієнти диктують вимоги згідно з умовами експлуатації	Обмеження для роботи на ринку через товари-замінники

Таблиця 6.9. (продовження)

На основі аналізу конкуренції, проведеного в п. 3.5 (таблиця 6.9), а також із урахуванням характеристик ідеї проекту (таблиця 6.2), вимог споживачів до товару (таблиця 6.5) та факторів маркетингового середовища (таблиця 6.6-6.7)

визначається та обґрунтовується перелік факторів конкурентоспроможності. Аналіз оформлюється за таблицею 10.

Таблиця 6.10. Обґрунтування факторів конкурентоспроможності

№ п/п	Фактор конкурентоспроможності	Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)
1	Відкритий програмний код	Існуючі конкуренти не надають доступ до коду своїх продуктів
2	Можливість розширення правил до декількох мов	Існуючі конкуренти надають аналіз тільки для однієї мови.

За визначеними факторами конкурентоспроможності (таблиця 6.10) проводиться аналіз сильних та слабких сторін стартап-проекту (таблиця 6.11).

Таблиця 6.11. Порівняльний аналіз сильних та слабких сторін

№ п/п	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з CodeAnalyzer (даним продуктом)				
			-2	-1	0	1	2
1	Широкий спектр правил	10		+			
2	Можливість модифікації шаблонів	20	+				

Фінальним етапом ринкового аналізу можливостей впровадження проекту є складання SWOT-аналізу (матриці аналізу сильних (Strength) та слабких (Weak) сторін, загроз (Troubles) та можливостей (Opportunities) (таблиця 6.12) на основі виділених ринкових загроз та можливостей, та

сильних і слабких сторін (таблиця 6.11).

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 6.12. SWOT-аналіз стартап-проекту

Сильні сторони: Можливість модифікації шаблонів Широкий спектр правил	Слабкі сторони: Відсутність графічного редактора
Можливості: Популярність мови програмування C# Відсутність повноцінних альтернатив	Загрози: Підходить для нових проектів Обмеженість функцій

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Ринкові загрози та ринкові можливості є наслідками (прогнозованими результатами) впливу факторів, і, на відміну від них, ще не є реалізованими на ринку та мають певну ймовірність здійснення.

Таблиця 6.13. Альтернативи ринкового впровадження стартап-проекту

№ п/ п	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Орієнтація поточної моделі на ринок стартаперів	50 %	40 год

2	Орієнтація поточної моделі на ринок державних установ	20 %	160 год
3	Орієнтація поточної моделі на ринок ентерпрайз	10 %	200 год
4	Переорієнтація на генерацію серверної частини	80 %	120 год
5	Переорієнтація на веб-розробку	35 %	80 год

Альтернатива, де отримання ресурсів є більш простим та ймовірним – №4 "Переорієнтація на генерацію серверної частини", що становить 80 відсотків. Це значення перевищує інші альтернативи.

Альтернатива, де строки реалізації є більш стислими – №1 "Орієнтація поточної моделі на ринок стартаперів". Терміни реалізації в цьому разі становлять лише 40 годин.

6.4 Розроблення ринкової стратегії проекту

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів (таблиця 6.14).

За результатами аналізу потенційних груп споживачів (сегментів) автори ідеї обирають цільові групи, для яких вони пропонуватимуть свій товар, та визначають стратегію охоплення ринку.

Таблиця 6.14. Вибір цільових груп потенційних споживачів

№ п/п	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Стартапери	Готові	Високий	Середня	Просто
2	Державні установи	Потребують недовгих переговорів	Середній	Середня	Складно
3	Ентерпрайз	Потребують довгих переговорів	Високий	Висока	Складно
Які цільові групи обрано: стартапери					

Для роботи в обраних сегментах ринку необхідно сформувати базову стратегію розвитку (таблиця 6.15).

Таблиця 6.15. Визначення базової стратегії розвитку

№ п/п	Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку*
1	Орієнтація поточної моделі на ринок стартаперів	Стратегія концентрованого маркетингу	Стартапери потребують швидкості розробки, яку надає підтримка декількох платформ даним продуктом	Стратегія спеціалізації (спирається на диференціацію)

Таблиця 6.15. (продовження)

Розроблення ринкової стратегії першим кроком передбачає визначення стратегії охоплення ринку: опис цільових груп потенційних споживачів.

Перелік ринкових загроз та ринкових можливостей складається на основі аналізу факторів загроз та факторів можливостей маркетингового середовища. Після визначення потенційних груп клієнтів проводиться аналіз ринкового середовища: складаються таблиці факторів, що сприяють ринковому впровадженню проекту.

Наступним кроком є вибір стратегії конкурентної поведінки (таблиця 6.16).

Таблиця 6.16. Визначення базової стратегії конкурентної поведінки

№ п/п	Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки*
1	Ні	шукати нових споживачів	Так, розділення на генерований код та код вільний до редагування	Стратегія заняття конкурентної ніші

З обраних сегментів до постачальника (стартап-компанії) та до продукту розробляється стратегія позиціонування (таблиця 6.17). що полягає у формуванні ринкової позиції, за яким споживачі мають ідентифікувати торгівельну марку/проект.

Таблиця 6.17. Визначення стратегії позиціонування

No п/п	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкуренто-спроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформулювати комплексну позицію власного проекту (три ключових)
1	Стабільність роботи Невисока ціна Наявність пробного періоду Наявність документації	Стратегія спеціалізації (спирається на диференціацію)	Стартапери потребують швидкості розробки якісного ПЗ	Пошук помилок в коді Рефакторинг C# .Net Гнучка система правил

6.5 Розроблення маркетингової програми стартап-проекту

Для цього у таблиці 6.18 потрібно підсумувати результати попереднього аналізу конкурентоспроможності товару.

Таблиця 6.18. Визначення ключових переваг концепції потенційного товару

No п/п	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Пришвидшення розробки ПЗ	Рекомендації по виправленню помилок	Автоматична корекція помилок
2	Підвищення якості ПЗ	Автоматичний пошук дефектів	Широкий спектр правил

Надалі розробляється трирівнева маркетингова модель товару:

уточняється ідея продукту та/або послуги, його фізичні складові, особливості процесу його надання (таблиця 6.19).

Таблиця 6.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
I. Товар за задумом	Генератор нативного модельного коду для мобільних платформ на основі заданої схеми БД		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	можливість оптимізації витрат часу	М	Тл
	можливість оптимізації витрат коштів	М	Вр
	відповідність актуальним технологіям	Нм	Тх
	Відповідає вимогам ДСТУ ISO/IEC 25030:2015 Програмна інженерія. Вимоги щодо якості та оцінювання програмного продукту (SQuaRE). Вимоги щодо якості		
	Пакування: готовий до використання dmg пакет		
	Марка: CodeAnalyzer		
III. Товар із підкріпленням	Потенційний користувач може ознайомитись з поточним товаром з наукових конференцій та публічних виступів, а також наукових вісників на яких була представлена інформація про даний продукт		
За рахунок чого потенційний товар буде захищено від копіювання: Назва і контент захищені ліцензією MIT; захист інтелектуальної власності			

Таблиця 6.19. (продовження)

М/Нм – монотонні або немонотонні;

Вр/Тх/Тл/Е/Ор – вартісні, технічні, технологічні, ергономічні або

органолептичні (останній – для продуктів харчування)

Після формування маркетингової моделі товару слід особливо відмітити – чим саме проєкт буде захищено від копіювання. Захист може бути організовано за рахунок захисту ідеї товару (захист інтелектуальної власності), або ноу-хау.

Наступним кроком є визначення цінових меж, якими необхідно керуватись при встановленні ціни на потенційний товар (таблиця 6.20).

Таблиця 6.20. Визначення меж встановлення ціни

№ п/п	Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
1	103...1500 \$	103...1500 \$	1000\$...5000 \$	103...1500 \$

Наступним кроком є визначення оптимальної системи збуту, в межах якого приймається рішення (таблиця 6.21):

- проводити збут власними силами або залучати сторонніх посередників (власна або залучена система збуту);
- вибір та обґрунтування оптимальної глибини каналу збуту;
- вибір та обґрунтування виду посередників.

Таблиця 6.21. Формування системи збуту

№ п/п	Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
--------------	--	--	-----------------------------	---------------------------------

1	Клієнт повинен надаватися в режимах “тріал” та “повний”. Сплата має проходити через AppStore	Легість в встановленні, легкість в сплаті послуг	4: Розробник даного продукту - Веб-сайт - App Store - Користувач.	Проводити збут силами посередника App Store
---	---	---	---	---

Останньою складовою маркетингової програми є розроблення концепції маркетингових комунікацій, що спирається на попередньо обрану основу для позиціонування, визначену специфіку поведінки клієнтів (таблиця 6.22).

Таблиця 6.22. Концепція маркетингових комунікацій

№ п/п	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Купляють програми через авторизовану мережу App Store	App Store, веб-сайти	підтримка декількох платформ Пришвидшення розробки ПЗ	Довести, що програмний продукт пришвидшить розробку ПЗ вразі	Один раз моделюй БД - генеруй код для декількох платформ

Таблиця 6.22. (продовження)

Висновки до розділу 6

Розроблений програмний продукт має переваги над існуючими конкурентами та є конкурентноздатним на ринку. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії та шляхи збуту. Для успішного виконання проекту необхідно реалізувати сервіс із використанням нейронних мереж для класифікації зображень. В рамках даного дослідження

були розраховані основні фінансово-економічні показники проекту, а також проведений менеджмент потенційних ризиків. Проаналізувавши отримані результати, можна зробити висновок, що подальша імплементація є доцільною.

В рамках даного дослідження були розраховані основні фінансово-економічні показники проекту, а також проведений менеджмент потенційних ризиків. Було визначено такі сильні сторони: форма виконання, собівартість та наявність адміністратора для налаштування. Натомість як слабкою є - крос-платформеність.

Наявні такі фактори загроз: конкуренція, зміна потреб користувачів, зміна тарифів провайдера, надходження на ринок альтернативних продуктів, уповільнення росту ринку.

ВИСНОВКИ

Після виконання магістрської дисертації можна зробити наступні висновки.

Обсяг інформації із звичайних баз даних та веб-джерел за останні кілька років експоненціально зростає. Як ефективно використовувати цей величезний обсяг інформації, стало проблемою для окремих людей та організацій. У той час як існуючий бізнес здійснює адаптаційні перетворення, щоб впоратися з динамікою цифрової економіки, з'являються нові технології та послуги, які допомагають інтегрувати та отримувати цінності з відповідних джерел інформації. Да даних момент існує велика кількість інформаційних джерел різни за своєю природою. Тому створення даного програмного засобу надає змогу користувачу ефетивніше та зручніше збирати інформацію з різних джерел та сама тема є актуальною.

Тому було прийнято рішення розробити систему програмних засобів для пошуку, збереження та аналізу для наступної обробки агрегованих інформаційних джерел. Така система має дати аналітику чітку інформацію про тематику заявки відповідно до цілей пошуку.

Задачі, що вирішуються за доаомогоюданої системи:

- аналіз існуючих систем пошуку інформації;
- підключення до існуючої інфомаційної бази;
- вилучення інформаційної бази за запитом;
- аналіз вилученої інформації в залежності від запиту;
- надання впорядженної інформації запиту.

Продукт було розроблено використовуючи мову програмування Python 3.6, з використанням структурного шаблону MVC та технології NLTK. Клієнтська частина була побудована за допомогою Qt Designer, для зберігання

інформації була використана реляційна база даних MySQL. Сервером додатків виступає WSGI.

Використання клієнт-серверної архітектури та веб-орієнтованість забезпечили дуже високу доступність програмного засобу та можливість запровадити єдиний інтерфейс для взаємодії з продуктом, незалежно від операційної системи, встановленої на користувацькому пристрої та його програмного оточення.

Таким чином, існування практики, знання різних технологій для розробки клієнт-серверних додатків і взаємодії з третьою стороною послуг, а також є дуже актуальна в сучасному світі так, як накоплюється дуже багато різних інформаційних баз. Програма має шляхи подальшого розвитку, визначені маркетингові стратегії та шляхи збуту. Основна цільова аудиторія – це студенти, стартап-команди, науковці, наукові організації та установи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Караєва Н. А. Соціально-економічні проблеми управління станом довкілля / Н. В. Караєва, І. В. Сегеда. — К: НТУУ “КПІ”, 2008. — 52 с.
2. Флэнаган Д. JavaScript. Подробное руководство / Дэвид Флэнаган. — СПб: Символ-Плюс, 2012. — 1080 с.
3. Боровков А. И. Компьютерный инжиниринг / А. И. Боровков. — СПб: Изд-во Политехн. ун-та, 2012. — 93 с.
4. Портнякин И. Эффективные пользовательские интерфейсы / Иван Портнякин. — Санкт-Петербург: Лори, 2011. — 600 с.
5. Хорстманн К. PHP 7. Вводный курс / Кей Хорстманн. — М: Вильямс, 2014. — 208 с.
6. Бауэр К. Java Persistence API и Hibernate / К. Бауэр. — М.: ДМК Пресс, 2017. — 335 с.
7. Батюк А.Є., кравчук Г.Т. автоматизація технологічного процесу обробки інформації в сучасних умовах.
8. MySQL — руководство администратора. — М.: Вильямс, 2005. — 201 с.
9. Дунаев В.В. (X)HTML, скрипты и стили. Самое необходимое / В.В. Дунаев. — М.: БХВ-Петербург, 2009. — 52 с.
10. Фримен Э. Изучаем HTML, XHTML и CSS / Эрик Фримен, Элизабет Фримен. — СПб.: Питер, 2012. — 656 с.
11. Головкин А. Встроенные стили [Электронный ресурс] / А. Головкин . — Режим доступа: <http://xiper.net/learn/css/css-including/inline-styles>.

12. Aggarwal C.C. Optimized crossover for maximum independent set / C.C. Aggarwal, J.B. Orlin, R.P. Tai // Oper. Res. — V. 45 (1997). — P. 225-234.
13. Талманн Л. Обеспечение высокой доступности систем на основе MySQL / Л. Талманн, М. Киндал, Ч. Белл. — М.: Вильямс, 2012. — 624 с.
14. MySQL [Электронный ресурс]. — Режим доступа: <http://www.mysql.com/>
15. Методика оценки уровня конкурентоспособности промышленной продукции. — М.: Изд-во стандартов, 1984
16. Сидоров М.О., Костів М.М.. Метод створення ефективного стилю програмування // Інженерія програмного забезпечення – 2013. – № 3–4 (15–16) – С. 17–24.
17. Beautiful Soup 4 Python [Электронный ресурс] – Режим доступа до ресурсу: <http://www.pythonforbeginners.com/beautifulsoup/beautifulsoup-4-python>.
18. Практические методы поиска анализа информации [Электронный ресурс]. Режим доступа: distcons.ru/modules/internet/index_02.htm. — Заголовок з екрану.
19. Procedures Guide: High-Performance Procedures, Second Edition [Электронный ресурс]. Режим доступа: http://support.sas.com/documentation/cdl/en/prochp/66704/HTML/default/viewer.htm#prochp_hpbin_details02.htm .
20. Weight of evidence and Information Value using Python. [Электронный ресурс]. Режим доступа: <https://medium.com/@sundarstyles89/weight-of-evidence-and-information-value-using-python-6f05072e83eb>.

21. SWOT-аналіз [Електронний ресурс] // Навчальні матеріали онлайн – Режим доступу до ресурсу: <http://pidruchniki.com/1577111551903/marketing/swot-analiz>.
22. Модель п'яти сил конкуренції за М. Портером [Електронний ресурс] – Режим доступу до ресурсу: <http://stud.com.ua/45490/ekonomika/>.
23. Цибульов П. М. Управління інтелектуальною власністю : монографія/ Цибульов П. М., Чеботарьов В. П., Зінов В. Г. , Суїні Ю., за ред. П. М. Цибульова. – К. : «К. І. С.», 2005. – 448 с.
24. Квашнин А. Как управлять портфелем технологий и интеллектуальной собственностью : серия методических материалов «Практические руководства для центров коммерциализации технологий» / под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. – 60 с.
25. Wagner B. Effective C# // Addison-Wesley Professional – 2016. ISBN: 9780134579290
26. Квашнин А. Как продвигать проекты коммерциализации технологий : серия методических материалов «Практические руководства для центров коммерциализации технологий» / М. Катешова, А. Квашнин, под рук. П. Линдхольма, проект EuropeAid «Наука и коммерциализация технологий», 2006. – 52 с.
27. Петруненко А. Оценка коммерческой привлекательности проекта [Електронний ресурс] // Технологический бизнес. – 1999. – № 2. Режим доступа: <http://www.techbusiness.ru/tb/archiv/number2/page01.htm>
28. Тиль, П. От нуля к единице : как создать стартап, который изменит будущее / П. Тиль, Б. Мастерс; перевод с англ. – Москва : Альпина паблишер, 2015. – 188 с.

- 29.Харниш, В. Правила прибыльных стартапов : как расти и зарабатывать деньги / В. Харниш ; пер. с англ. В. Хозинского. – Москва : Манн, Иванов и Фербер, 2012. – 279 с
- 30.Экланд С. Ангелы, драконы и стервятники : как привлечь правильных инвесторов в свой стартап и сохранить бизнес / С. Экланд ; пер. с англ. О. Терентьевой. – Москва : Манн, Иванов и Фербер, 2011. – 275 с.
- 31.Маллинс, Дж. Поиск бизнес-модели : как спасти стартап, вовремя сменив план / Дж. Маллинс, Р. Комисар ; пер. с англ. М. Пуксант и Е. Бакушевой. – Москва : Манн, Иванов и Фербер, 2012. – 329 с.
- 32.Clarke-Salt J. SQL Injection Attacks and Defense / J. Clarke-Salt. Syngress – 2009, pp. 153-154. ISBN: 9781597499736
- 33.The Ten Most Critical Web Application Security Risks [Электронный ресурс] // OWASP – Режим доступа до ресурсу: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf
- 34.Meszaros G. xUnit Test Patterns: Refactoring Test Code / G. Meszaros. - Addison-Wesley Professional, 2007. – p. 311. ISBN: 9780131495050